Contributions to Structural Communication Complexity

# Contributions to Structural Communication Complexity

Dissertation zur Erlangung des akademischen Grades
Doctor rerum naturalium (Dr. rer. nat.)

vorgelegt der Fakultät für Informatik und Automatisierung
der Technischen Universität Ilmenau

von

Dipl.-Inf. Henning Wunderlich

Vorgelegt am: 25.05.2009

Gutachter:

1. Univ.-Prof. Dr.(USA) habil. Martin Dietzfelbinger,
   Technische Universität Ilmenau
2. Univ.-Prof. Dr. rer. nat. habil. Georg Schnitger,
   Goethe-Universität Frankfurt
3. Univ.-Prof. Dr. rer. nat. habil. Uwe Schöning,
   Universität Ulm

urn:nbn:de:gbv:ilm1-2009000312

dedicated to Siegrid Sygusch

# Contents

# List of Figures

# List of Tables

# Deutsche Zusammenfassung

Neben Turing-Maschinen und Schaltungen ist eines der wichtigsten und interessantesten Berechnungsmodelle Yaos Kommunikationsmodell. Dort wird mittels Protokollen modelliert, wie zwei Spieler durch Kommunikation gemeinsam ein Problem lösen, für welches sie nur einen Teil der Eingabe besitzen. Eine grundlegende Fragestellung ist es, bei einem gegebenen Problem zu bestimmen, wieviele Bits kommuniziert werden müssen, um es zu lösen. In Kapitel 1 geben wir eine kurze Einführung in die Kommunikationskomplexitätstheorie, in der Kosten- und Komplexitätsbegriffe eingeführt werden, um diese Fragestellung zu untersuchen. Zu jedem Berechnungsmodell existiert eine Strukturelle Komplexitätstheorie, insbesondere auch zu Yaos Modell, in der Aussagen über Familien von Problemen gemacht werden. Die vorliegende Dissertation enthält Beiträge zum Gebiet der Strukturellen Kommunikationskomplexitätstheorie, die sich thematisch drei Bereichen dieses Gebietes zuordnen lassen:

Der erste Teil, Kapitel 4, beschäftigt sich mit dem fundamentalen Problem der Herleitung unterer Schranken für die randomisierte Kommunikationskomplexität. Eine dafür wichtige Technik ist die Anwendung informationstheoretischer Methoden auf die durchschnittliche deterministische Kommunikationskomplexität. Dort betrachtet man Protokolle, die eine Funktion berechnen, die aber möglichst wenig Information über die Eingabe preisgeben. Gemessen wird dies durch Begriffe von Informationskomplexitäten. Unser Hauptergebnis ist folgende Charakterisierung, die auch als nicht-triviale Verallgemeinerung von Shannons „Noiseless Coding Theorem“ angesehen werden kann: die durchschnittliche deterministische Informationskomplexität stimmt bis auf einen konstanten Faktor mit der durchschnittlichen deterministischen Kommunikationskomplexität überein, in Formeln

$$\mathrm{D}_0^{\mu}(f) = \Theta\left(\mathrm{IC}_0^{\mathrm{det},\mu}(f)\right) \ ,$$

und zwar für jede Funktion $f$ und jede Verteilung $\mu$ auf den Eingaben. Ein weiteres Ergebnis sind untere Schranken für die durchschnittliche randomisierte public coin Kommunikationskomplexität, die um einen konstanten Faktor besser sind, als die bis dahin bekannten.

Der zweite Teil der Arbeit, Kapitel 5, beschäftigt sich mit der seit über 30 Jahre alten Fragestellung, ob in der Kommunikationskomplexität die Polynomielle Hierarchie $\mathbf{PH}^{cc}$ eine echte Teilmenge des Polynomiellen Platzes $\mathbf{PSPACE}^{cc}$ ist. Diese Klassen sind kommunikationskomplexitätstheoretische Analoga der Polynomiellen Hierarchie und des Polynomiellen Platzes aus dem Turing-Maschinenmodell. Die Fragestellung wird hier nicht gelöst, aber es wird eine Richtung aufgezeigt, wie dieses Problem vielleicht gelöst werden könnte. Eine Schwierigkeit, die beiden auf Alternierungen basierenden Klassen zu trennen, besteht darin, dass keine Maße für Alternierungen existieren. Wir übertragen deshalb in einem ersten Schritt die Todaschen Sätze aus der klassischen Komplexitätstheorie in Yaos Modell. Der für uns wichtige erste Todasche Satz besagt, dass zwischen den beiden Klassen die Klasse BP-Parität-P liegt:

$$\mathbf{PH}^{cc} \subseteq \mathrm{BP} \cdot \oplus\mathbf{P}^{cc} \subseteq \mathbf{PSPACE}^{cc} \ .$$

In einem zweiten Schritt entwickeln wir ein Maß für die Klasse BP-Parität-P, nämlich den approximativen $\mathbb{F}_2$-Rang, für den wir eine enge Beziehung mit einer Booleschen Variante des Konzepts „matrix rigidity“ herstellen. Somit liefern untere Schranken für die Boolesche matrix rigidity untere Schranken für die Kommunikationskomplexität.

Wir erhalten dadurch und durch Benutzung eines Resultates von Valiant ein Maßkonzentrationsresultat für die BP-Parität-P-Komplexität: die meisten Funktionen haben eine BP-Parität-P-Komplexität von $\Omega(n/\log n)$. Wir entwickeln ein Protokoll für die innere Produktfunktion mod 2 mit wenigen Alternierungen. Dies könnte darauf hinweisen, dass die Klassen BP-Parität-P und Polynomieller Platz verschieden sind. Wir denken, dass Adjazenzprobleme, die auf dünnen quasi-zufälligen Graphfamilien basieren, eine hohe BP-Parität-P-Komplexität besitzen. Wir können dies nicht beweisen, aber wir können zumindest zeigen, dass für die Parität-P-Komplexität einer quasizufälligen Graphfamilie $(G_n)_{n\in\mathbb{N}}$ gilt:

$$\oplus\mathrm{P}(\mathrm{EDGE}_{G_n}) \geq \log\left(\frac{1}{P(n)}\right) - \mathcal{O}(1) \ ,$$

wobei $P(n)$ die Kantendichte der Graphfamilie bezeichnet.

Im dritten und damit letzten Teil, Kapitel 6, untersuchen wir, motiviert durch ein seit langer Zeit offenes Problem in der Kommunikationskomplexität, Überdeckungsstrukturgraphen. Diese sind definiert als Schnittgraphen von maximalen monochromatischen Untermatrizen einer Matrix. Wir zeigen, dass nicht jeder Graph ein Überdeckungsstrukturgraph ist, darunter fallen insbesondere Quadrate und ungerade Löcher (induzierte Kreise ungerader Länge). Es ist ganz natürlich, Graphen zu betrachten – wir nennen sie „schöne“ Graphen –, die die Eigenschaft haben, dass jeder induzierte Untergraph ein Überdeckungsstrukturgraph ist. Per definitionem bilden sie eine neue, sehr spezielle Klasse von quadratfreien Berge-Graphen. Ein tiefliegender Satz von Conforti *et al.* besagt, dass sich jeder quadratfreie Berge-Graph $G$ folgendermaßen zerlegen lässt: $G$ ist bipartit oder ein Kantengraph eines bipartiten Graphen, oder er enthält eine Stern-Schnittmenge (star cutset) oder einen 2-Verbund (2-join).

Wir machen Fortschritte hinsichtlich einer analogen spezielleren Zerlegung für schöne Graphen: jeder bipartite Graph ist schön, und die schönen Kantengraphen bipartiter Graphen sind genau die „Path-or-Even-Cycle-of-Cliques“-Graphen, d. h. diejenigen Graphen, die aus Wegen beliebiger und Kreisen gerader Länge bestehen, an deren Kanten Cliquen beliebiger Größe angeheftet sein können.

# Acknowledgments

First and foremost, I would like to express my deep gratitude towards my adviser and educator Martin Dietzfelbinger. I learned such a lot from him, especially through the many fruitful and stimulating discussions we had each time after he carefully and patiently proof-read the numerous trials I made as a beginning researcher. He is one of the very few people I would consider to belong to Germany's elite in the sense of national economist Wilhelm Röpke:

> "Eine wahre Elite würde eine Stellung über den Klassen, Interessen, Leidenschaften, Bosheiten und Torheiten der Menschen einnehmen. Sie würde sich auszeichnen durch ein exemplarisches und langsam reifendes Leben der entsagungsvollen Leistung für das Ganze, der unantastbaren Integrität und (...) durch unerschütterlichen Mut im Eintreten für das Wahre und Rechte. "

A change of air is always good to inspire new ideas. I am extraordinarily indebted to Uwe Schöning and Jacobo Torán for offering me a position as an "Akademischer Mitarbeiter" at Ulm University and for providing me a stimulating environment I benefitted so much from during the last couple of years.

Furthermore I offer 1024*1024 thanks to Michael Brinkmeier, Elke Hübel, Manfred Kunde, Karl-Heinz Niggl, Enno Ohlebusch, Andre Osterloh, Ulf Schellbach, Thomas Thierauf and all former and current members of the teams "Theoretische Informatik" at Technical University Ilmenau and Ulm University for being such a welcoming group of people.

Especially, I would like to thank Karl-Heinz Niggl who encouraged me to start doing research in theoretical computer science and supported me with wise advice. He is a true friend.

I would like to thank Hartmut Klauck for visiting me in Ulm giving me insight into his state of the art research.

Special thanks go to Andreas Brandstädt, Thanh Minh Hoang, Michael Stiebitz and Fabian Wagner for their careful reading of the graph theory paper that constitutes the third part in this thesis and their helpful comments and suggestions that have led to substantial improvements.

In addition, I would like to thank an anonymous referee of the graph theory paper mentioned above, who provided an alternative proof sketch of the characterization of beautiful line graphs of bipartite square-free graphs. The respective proof sketch has been worked out in detail and is included in this thesis.

Finally, I would like to sincerely thank my parents for all the support and encouragement they have given me over the years!

# List of publications

Parts of this thesis are based on the following papers:

1. Martin Dietzfelbinger & Henning Wunderlich (2007). A characterization of average case communication complexity. *Inf. Process. Lett.* **101**(6), 245–249.

2. Henning Wunderlich (2009b). On Toda's Theorem in Structural Communication Complexity. In *SOFSEM 2009: Theory and Practice of Computer Science, 35th Conference on Current Trends in Theory and Practice of Computer Science, Spindleruv Mlýn, Czech Republic, January 24–30, 2009. Proceedings*, Mogens Nielsen, Antonín Kucera, Peter Bro Miltersen, Catuscia Palamidessi, Petr Tuma & Frank D. Valencia, editors, volume 5404 of *Lecture Notes in Computer Science*, 609–620. Springer-Verlag.

3. Henning Wunderlich (2009a). On cover-structure graphs, *Discrete Applied Mathematics* **157**(15), 3289–3299.

# 1 Summary

This thesis contains contributions to the field of structural communication complexity. As the name suggests we are interested in statements about families of problems concerning their communication complexity. An introduction to the communication model and the communication complexity measures considered in this thesis is given in Chapter 3. For notation used throughout this thesis, see Chapter 2.

Thematically, this work can be divided into three parts:

## 1.1 Information complexity

The first part, Chapter 4, is based on the publication "A characterization of average case communication complexity" (Dietzfelbinger & Wunderlich 2007). A fundamental problem is the derivation of lower bounds for randomized communication complexity. An important technique that has led to striking results is the application of information-theoretical methods on average-case deterministic communication complexity. There, one considers protocols that compute a function but only reveal little about the inputs. Quantitatively, this is measured via notions of information complexities. Our main result is the following characterization, which may be considered as a non-trivial generalization of Shannon's Noiseless Coding Theorem: average case deterministic communication complexity and average case deterministic information complexity only differ by a constant factor, i. e.

$$\mathrm{D}_0^{\mu}(f) = \Theta\left(\mathrm{IC}_0^{\mathrm{det},\mu}(f)\right) \ ,$$

for every function $f$ and every probability distribution $\mu$ on the inputs. We also note improved lower bounds for average case randomized public coin communication complexity that are a constant factor better than known previous bounds.

## 1.2 Structural communication complexity

The second part, Chapter 5, deals with the over thirty year old question, whether the polynomial hierarchy $\mathbf{PH}^{cc}$ is a proper subset of polynomial space $\mathbf{PSPACE}^{cc}$. The respective classes are analogues of the polynomial hierarchy and polynomial space in the Turing machine model. We do not solve this problem here, but we discuss a possible promising direction to settle it. The first difficulty one encounters when one tries to separate the polynomial hierarchy from polynomial space is that both classes are based on the concept of alternation but one does not have measures for alternation. In a first step we translate Toda's celebrated theorems to the setting of communication complexity. This result was presented in "On Toda's Theorem in Structural Communication Complexity" (Wunderlich 2009b). In particular, Toda's First Theorem tells us that the complexity class BP-Parity-P, $\mathrm{BP} \cdot \oplus\mathbf{P}^{cc}$, is in a "sandwich" position between the two alternating classes:

$$\mathbf{PH}^{cc} \subseteq \mathrm{BP} \cdot \oplus\mathbf{P}^{cc} \subseteq \mathbf{PSPACE}^{cc} \ .$$

Now, the second step consists in deriving a measure that characterizes the class BP-Parity-P. It turns out that the respective measure is (the logarithm of) approximate $\mathbb{F}_2$-rank. In addition, we show a tight connection between a Boolean variant of the

concept "matrix rigidity". Thus, lower bounds for Boolean matrix rigidity yield lower bounds for communication complexity. Via a result of Valiant we obtain a concentration of measure result for BP-Parity-P complexity: most functions have complexity $\Omega(n/\log n)$. We think that adjacency problems of sparse quasi-random graph-families have high BP-Parity-P complexity. Unfortunately, we cannot prove this, but at least we can show that for a quasi-random graph family $(G_n)_{n\in\mathbb{N}}$ we have

$$\oplus\mathrm{P}(\mathrm{EDGE}_{G_n}) \geq \log\left(\frac{1}{P(n)}\right) - \mathcal{O}(1) \ ,$$

where $P(n)$ denotes the edge density of $G_n$.

## 1.3 Cover-structure graphs

The third and last part of this thesis, Chapter 6, is based on the paper "On cover-structure graphs" (Wunderlich 2009a). Motivated by a long-standing open problem in communication complexity we study *cover-structure graphs (cs-graphs)* defined as intersection graphs of maximal monochromatic submatrices in a matrix. We show that not every graph is a cs-graph. Especially, squares and odd holes are not cs-graphs. It is natural to look at graphs – we call them *beautiful graphs* – having the property that each induced subgraph is a cs-graph. By definition, they form a new very special class of square-free Berge graphs. A deep result of Conforti *et al.* tells us that every square-free Berge graph $G$ can be decomposed in the following way: $G$ is bipartite or the line graph of a bipartite graph, or $G$ contains a star cutset or a 2-join. We make progress towards an analogous but more special decomposition by showing that every square-free bipartite graph is beautiful, and that beautiful line graphs of square-free bipartite graphs are just *Path-or-Even-Cycle-of-Cliques* graphs. The latter are graphs that consist of paths of arbitrary length or cycles of even length, where to each edge there may be attached a clique of arbitrary size.

# 2 Preliminaries

We fix notation and give basic definitions used throughout this thesis.

We denote with $[n]$ the set $\{1, \ldots, n\}$ of the first $n$ natural numbers. We write $\mathcal{P}(S)$ for the *power set of* $S$, i. e. the set of all subsets of $S$, and $\binom{S}{k}$ for the set of all subsets of $S$ with cardinality $k$.

For a real number $r$ we denote with *floor* $r$, $\lfloor r \rfloor$, the largest integer not exceeding $r$, and with *ceiling* $r$, $\lceil r \rceil$, the smallest integer greater than or equal to $r$.

The logarithm to the basis 2 is denoted with log.

For a prime power $p$ we denote with $\mathbb{F}_p$ the finite field with $p$ elements.

We define matrices with arbitrary finite index sets for rows and columns. Thus, a matrix $M$ over $\mathcal{Z}$ is just a map $M\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ for finite sets $\mathcal{X}$ and $\mathcal{Y}$. We write $M_{x,y}$ for $M$'s entry in row $x \in \mathcal{X}$ and column $y \in \mathcal{Y}$. Given two matrices $M_i\colon \mathcal{X}_i \times \mathcal{Y}_i \to \mathcal{Z}_i$, $i \in [2]$, we define the *block diagonal matrix* $\operatorname{diag}(M_1, M_2)\colon (\mathcal{X}_1 \uplus \mathcal{X}_2) \times (\mathcal{Y}_1 \uplus \mathcal{Y}_2) \to (\mathcal{Z}_1 \cup \mathcal{Z}_2 \cup \{0\})$ as

$$\operatorname{diag}(M_1, M_2) := \begin{pmatrix} M_1 & 0 \\ 0 & M_2 \end{pmatrix} .$$

Occasionally, in order to avoid ugly case distinctions we use *Iverson's bracket* $[P]$ defined on predicates $P$, which evaluates to 1, if $P$ is true, and 0 otherwise.

We only work with the *binary alphabet* $\mathbb{B} := \{0, 1\}$. The length of a word $x \in \mathbb{B}^*$ is denoted by $|x|$. For two words $x, y \in \mathbb{B}^*$ the word $y$ is a *prefix of* $x$, if there exists $z \in \mathbb{B}^*$ such that $x = yz$. A set $S \subseteq \mathbb{B}^*$ is *prefix-free* if for all distinct $x, y \in S$ we have that $y$ is not a prefix of $x$. A *prefix-free encoding* of $x$ is $\overline{x} := 0^{|x|}1x$. In order to encode pairs of words $x, y \in \mathbb{B}^*$ we use the *pairing function* $\langle x, y \rangle := \overline{x}y$. For a mathematical object $o$ contained in an at most countable set we denote with $\langle o \rangle$ a suitable prefix-free encoding of $o$.

Functions with range $\mathbb{B}$ are called *Boolean functions*.

For an excellent introduction to graph theory we refer the reader to Diestel (2005). Given a *graph* $G$ we write $V(G)$ to denote its *nodes* (*vertices*) and $E(G)$ to denote its *edges*. A set $K \subseteq V(G)$ is a *complete set* in $G$, if $G$ contains all edges between nodes in $K$. A set $I \subseteq V(G)$ is an *independent set* in $G$, if $G$ does not contain any edges between nodes in $I$. The *complement graph* of $G$ is defined as $\overline{G} := (V(G), \overline{E})$, where $\overline{E}$ contains exactly the edges not in $E(G)$. Let $G_1$ and $G_2$ be two graphs with disjoint node sets. We define their *disjoint union* as $G_1 \biguplus G_2 := (V(G_1) \uplus V(G_2), E(G_1) \uplus E(G_2))$. A 4-cycle $C_4$ is called a *square*; a *square-free* graph does not contain a square as an induced subgraph. The *line graph* of the graph $G$ is the graph $L(G)$ whose nodes are the edges of $G$ and two nodes $u$, $v$ of $L(G)$ are adjacent in $L(G)$ iff the edges $u$, $v$ of $G$ are incident to a common node of $G$. We write $G =_{\text{iso}} H$ iff $G$ and $H$ are isomorphic, and $G \leq_{\text{iso}} H$ iff $G$ is isomorphic to an induced subgraph of $H$. As usual, the *adjacency matrix of* $G$, $A^G$, is defined by

$$A^G_{x,y} := [\{x, y\} \in E(G)] \quad \text{, for } x, y \in V(G) \ .$$

For subsets $X, Y \subseteq V(G)$ we define the *set of edges between* $X$ *and* $Y$ as

$$E_G(X, Y) := \{\{x, y\} \mid x \in X, y \in Y\} \ .$$

We also define $e_G(X, Y)$ as the number of edges with one endpoint in $X$ and the other one in $Y$. If an edge belongs to the intersection $X \cap Y$, then it is counted twice in $e_G(X, Y)$.

# 3 Communication complexity

In this chapter we give a short introduction to (parts of) communication complexity. In particular, we describe Yao's model and state some important basic results that are used in later chapters. We refer the reader to Kushilevitz & Nisan (1997) for an excellent introduction to the field of communication complexity.

## 3.1 Yao's model

In this section we describe the communication model under consideration in this thesis.

### 3.1.1 Deterministic protocols

In his seminal work, Yao (1979) introduced a simple communication model. In *Yao's model*, there are two players (parties) Alice and Bob with unlimited computational power, who want to cooperatively compute a function $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, where $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ are finite sets. Both have complete information about $f$ but receive only parts of the input. Alice is given $x \in \mathcal{X}$, Bob is given $y \in \mathcal{Y}$, and they exchange messages (bits) in order to compute $f(x, y)$. The players communicate according to a fixed *(deterministic) protocol* $\Pi$ (over domain $\mathcal{X} \times \mathcal{Y}$ with range $\mathcal{Z}$) that specifies how the communication is carried out. At each stage of the computation, the protocol must determine whether the run has terminated. In this case, it must specify the output value. Otherwise, it must specify the player who speaks next. Each message sent by a player must solely depend on the player's input and the messages communicated so far, because this is the only "information" the player has about the inputs.

While Yao only considered players computing *functions* the generalization to the case of *relations* was initiated by Karchmer & Wigderson (1990) motivated by applications in the theory of Boolean functions. In this case, on input $(x, y)$ there might be none or several valid outputs. A *relation* is just a subset $R \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. For a *legal input* $(x, y)$, i. e. an input such that there exists a value $z \in \mathcal{Z}$ with $(x, y, z) \in R$, the players want to compute a value $z \in \mathcal{Z}$ such that $(x, y, z) \in R$. In this extension to Yao's model the notion of "protocol" remains unchanged, and the associated cost and complexity measures are only slight extensions of the ones defined for functions. This is why we do not write them down formally.

There exist different formalizations of the notion "protocol" depending on the applications the protocol designer has in mind. Since we want to analyze protocols, we formalize them via binary trees. See e. g. Hromkovic (2000) for a definition that is equivalent but different from ours. It is known that our combinatorial view on protocols has many advantages. In particular, it allows us to prove high lower complexity bounds in this model in contrast to many other computation models. Now, we formally define protocols:

Definition 3.1.1 (Deterministic protocol). *A* deterministic protocol $\Pi$ (over domain $\mathcal{X} \times \mathcal{Y}$ with range $\mathcal{Z}$) *is a labeled directed finite binary tree (*protocol tree*). Each leaf $l$ is labeled by an output value $z_l \in \mathcal{Z}$. If $v$ is an inner node of $\Pi$, then it has a left and a right child $v_0$ and $v_1$, respectively, and the arc from $v$ to $v_b$ is labeled by $b \in \{0, 1\}$. The node $v$ is labeled either by a function $a_v\colon \mathcal{X} \to \{0, 1\}$ or by a function $b_v\colon \mathcal{Y} \to \{0, 1\}$. The root of the protocol tree of $\Pi$ is denoted by* $\mathrm{root}(\Pi)$*, the set of nodes by* $\mathrm{V}_\Pi$*, the set of leaves by* $\mathrm{L}_\Pi$*.*

Let Alice have $x \in \mathcal{X}$, and let Bob have $y \in \mathcal{Y}$. When they communicate according to a protocol $\Pi$, they start at the root $\mathrm{root}(\Pi)$. The nodes of the protocol tree of $\Pi$ can be interpreted as (common) "computation states". If both players are in such a state $v$ during the run of the protocol, then one of two things can happen: If $v$ is a leaf, then the communication ends and both players know the output value $z_v$. If $v$ is an inner node, we say that *Alice speaks*, if $v$ is labeled by $a_v$. Alice sends the bit $b := a_v(x)$ and both players change their computation state to $v_b$. Analogously, if *Bob speaks*.

We say that an input $(x, y)$ *reaches* a node $v$ of $\Pi$ if the players arrive at $v$ when running the protocol on the respective input. We denote by $R_v \subseteq \mathcal{X} \times \mathcal{Y}$ the *set of inputs reaching* $v$.

The concatenation of the messages communicated during a run of a protocol $\Pi$ on input $(x, y)$ is called *transcript* and is denoted by $\Pi(x, y)$. This can be defined inductively as follows:

Definition 3.1.2 (Transcript). *Let $\Pi$ be a deterministic protocol. Given an input $(x, y)$, we associate with each node $v$ of $\Pi$ a* transcript *$t_v(x, y)$. If $l$ is a leaf in $\Pi$, then $t_l(x, y) := \epsilon$. If $v$ is an inner node of $\Pi$ with left and right child $v_0$ and $v_1$, respectively, and if $v$ is labeled by $a_v$, then $t_v(x, y) := bt_{v_b}(x, y)$, where $b := a_v(x)$. In case $v$ is labeled by $b_v$, then $t_v(x, y) := bt_{v_b}(x, y)$, where $b := b_v(y)$. We define the* transcript *$\Pi(x, y)$ of $\Pi$ on input $(x, y)$ as the transcript $t_{\mathrm{root}(\Pi)}(x, y)$ of the root.*

For each input $(x, y)$ the execution of a deterministic protocol $\Pi$ leads to exactly one output value. This defines a function $f_\Pi$. We give an inductive definition:

Definition 3.1.3 (Computed function). *Let $\Pi$ be a deterministic protocol over domain $\mathcal{X} \times \mathcal{Y}$ with range $\mathcal{Z}$. We associate a function $f_{\Pi,v} \colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ with each node $v$ of $\Pi$. For a leaf $l$ we define $f_{\Pi,l}$ to be constant with value $z_l$. In case $v$ is an inner node with left and right child $v_0$ and $v_1$, respectively, we define $f_{\Pi,v}(x, y) := f_{\Pi,v_{a_v(x)}}(x, y)$, if Alice speaks, and $f_{\Pi,v}(x, y) := f_{\Pi,v_{b_v(y)}}(x, y)$, if Bob speaks. We define the* function *$f_\Pi$ computed by $\Pi$ as the function $f_{\Pi,\mathrm{root}(\Pi)}$ associated with the root.*

*We say that $\Pi$* computes *a function $f \colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, if $f = f_\Pi$. The protocol $\Pi$* computes *a relation $R \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, if for each legal input $(x, y)$, we have $(x, y, f_\Pi(x, y)) \in R$.*

Example 3.1.4. Given a function $f$, the *trivial protocol* $\Pi^f_{\mathrm{triv}}$ is defined as the one where Alice sends her input, Bob computes the value $f(x, y)$ and sends it back to Alice. Of course, $f = f_{\Pi^f_{\mathrm{triv}}}$ and $\Pi(x, y) = \langle x \rangle \langle f(x, y) \rangle$ for suitable (prefix-free) encodings of the values sent.

In particular, we obtain $\mathrm{D}(\Pi^f_{\mathrm{triv}}) = n + 1$ for a Boolean function $f \colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$. $\Diamond$

Having defined a computation model and a resource it is time to define corresponding cost and complexity measures. We distinguish between two types of complexities, namely *worst* and *average case* ones. The latter depend on probability distributions on the inputs.

Definition 3.1.5 (Deterministic communication cost). *Let $\Pi$ be a deterministic protocol over domain $\mathcal{X} \times \mathcal{Y}$, let $\mu$ be a probability distribution on $\mathcal{X} \times \mathcal{Y}$, and let $(X, Y)$ be a random variable distributed according to $\mu$.*

(i) *The* worst case deterministic communication cost of $\Pi$, $\mathrm{D}(\Pi)$, *is defined as*

$$\mathrm{D}(\Pi) := \max_{(x,y) \in \mathcal{X} \times \mathcal{Y}} |\Pi(x, y)| \ .$$

(ii) *The* $\mu$-average case deterministic communication cost of $\Pi$, $\overline{\mathrm{D}}^{\mu}(\Pi)$, *is defined as*

$$\overline{\mathrm{D}}^{\mu}(\Pi) := \mathrm{E}_{(X,Y)}\left[\Pi(X,Y)\right] \ .$$

DEFINITION 3.1.6 (Closeness). *Let $f$ and $g$ be two functions defined on the same domain $D$, let $\mu$ be a probability distribution on $D$, and let $\epsilon \geq 0$ be a real number. The functions $f$ and $g$ are* $(\mu,\epsilon)$-close, *if $\mu(f \neq g) := \mu\{z \in D \mid f(z) \neq g(z)\} \leq \epsilon$.*

DEFINITION 3.1.7 (Distributional error). *Let $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ be a function, let $\mu$ be a probability distribution on $\mathcal{X} \times \mathcal{Y}$, let $\epsilon \geq 0$ be a real number, and let $\Pi$ be a deterministic protocol over domain $\mathcal{X} \times \mathcal{Y}$ with range $\mathcal{Z}$. We say that* $\Pi$ computes $f$ with $(\mu,\epsilon)$-distributional error, *if $f$ and $f_\Pi$ are $(\mu,\epsilon)$-close.*

DEFINITION 3.1.8 (Deterministic communication complexity). *Let $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ be a function, let $\mu$ be a probability distribution on $\mathcal{X} \times \mathcal{Y}$, and let $\epsilon > 0$ be a real number.*

(i) *The* worst case deterministic communication complexity of $f$, $\mathrm{D}(f)$, *is the minimum worst case deterministic communication cost of a deterministic protocol computing $f$, i. e.*

$$\mathrm{D}(f) := \min\{\mathrm{D}(\Pi) \mid \Pi \text{ a deterministic protocol computing } f\} \ .$$

(ii) *The* $\mu$-average case deterministic communication complexity of $f$, $\mathrm{D}^{\mu}_0(f)$, *is the infimum $\mu$-average case deterministic communication cost of a deterministic protocol computing $f$, i. e.*

$$\mathrm{D}^{\mu}_0(f) := \inf\{\overline{\mathrm{D}}^{\mu}(\Pi) \mid \Pi \text{ a deterministic protocol computing } f\} \ .$$

(iii) *The* worst case $(\mu,\epsilon)$-distributional deterministic communication complexity of $f$, $\mathrm{D}^{\mu}_{\epsilon}(f)$, *is the minimum worst case deterministic communication cost of a deterministic protocol computing $f$ with $(\mu,\epsilon)$-distributional error, i. e.*

$$\begin{aligned}\mathrm{D}^{\mu}_{\epsilon}(f) \quad := \quad & \min\{\mathrm{D}(\Pi) \mid \Pi \text{ a deterministic protocol computing } f \text{ with} \\ & (\mu,\epsilon)\text{-distributional error}\} \ .\end{aligned}$$

(iv) *The* $\mu$-average case $(\mu,\epsilon)$-distributional deterministic communication complexity of $f$, $\overline{\mathrm{D}}^{\mu}_{\epsilon}(f)$, *is the infimum $\mu$-average case deterministic communication cost of a deterministic protocol computing $f$ with $(\mu,\epsilon)$-distributional error, i. e.*

$$\begin{aligned}\overline{\mathrm{D}}^{\mu}_{\epsilon}(f) \quad := \quad & \inf\{\overline{\mathrm{D}}^{\mu}(\Pi) \mid \Pi \text{ a deterministic protocol computing } f \text{ with} \\ & (\mu,\epsilon)\text{-distributional error}\} \ .\end{aligned}$$

The following properties should be obvious:

OBSERVATION 3.1.9. *Let $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ be a function, let $\mu$ be a probability distribution on $\mathcal{X} \times \mathcal{Y}$, and let $\epsilon > 0$ be a real number. Then we have*

$$\begin{aligned}\mathrm{D}^{\mu}_0(f) &\leq \mathrm{D}(f) \ , \\ \mathrm{D}^{\mu}_{\epsilon}(f) &\leq \mathrm{D}(f) \ , \\ \overline{\mathrm{D}}^{\mu}_{\epsilon}(f) &\leq \mathrm{D}^{\mu}_{\epsilon}(f) \ .\end{aligned}$$

OBSERVATION 3.1.10. *Let $f\colon \mathcal{X}\times\mathcal{Y}\to\mathcal{Z}$ be a function, let $\mu$ be a probability distribution on $\mathcal{X}\times\mathcal{Y}$, and let $\epsilon>0$ be a real number. Then we have*

$$\begin{aligned}\mathrm{D}^{\mu}_{\epsilon}(f) &= \min\{\mathrm{D}(\tilde{f}) \mid \tilde{f} \text{ and } f \text{ are } (\mu,\epsilon)\text{-close}\}\ ,\\ \overline{\mathrm{D}}^{\mu}_{\epsilon}(f) &= \inf\{\mathrm{D}^{\mu}_{0}(\tilde{f}) \mid \tilde{f} \text{ and } f \text{ are } (\mu,\epsilon)\text{-close}\}\ .\end{aligned}$$

We note some basic continuity properties of the cost and complexity measures defined above. Therefore, let $\mathcal{M}=\mathcal{M}(\mathcal{X}\times\mathcal{Y})$ denote the set of probability distributions $\mu\colon \mathcal{X}\times\mathcal{Y}\to[0,1]$ over the input space $\mathcal{X}\times\mathcal{Y}$.

LEMMA 3.1.11. *The set $\mathcal{M}$ is a convex and compact set.*

PROOF. For the first statement, note that a convex combination $(1-t)\cdot\mu+t\cdot\nu$ of two probability distributions $\mu,\nu$ is again a probability distribution.

For the second statement, let $L(z):=||z||_1:=\sum_{i=1}^{|\mathcal{X}\times\mathcal{Y}|}|z_i|$ be the $l_1$-norm on $\mathbb{R}^{\mathcal{X}\times\mathcal{Y}}$. $L$ is continuous, so $L^{-1}(1)=L^{-1}(\{1\})$ is closed. The set $\mathcal{H}:=\{z\in\mathbb{R}^{\mathcal{X}\times\mathcal{Y}}\mid z\geq 0\}=[0,\infty[^{\mathcal{X}\times\mathcal{Y}}$ is closed, too. Thus, $\mathcal{M}=\mathcal{H}\cap L^{-1}(1)$ is a closed subset of $\mathbb{R}^{\mathcal{X}\times\mathcal{Y}}$. By definition of a probability distribution, $L(\mu)=1$ for every $\mu\in\mathcal{M}$. By the Theorem of Heine-Borel, $\mathcal{M}$ as a closed and bounded set is compact. □

LEMMA 3.1.12.

(i) *For fixed $\Pi$, the mapping $\mu\mapsto\overline{\mathrm{D}}^{\mu}(\Pi)$ is continuous.*

(ii) *For fixed $f$, the mapping $\mu\mapsto\mathrm{D}^{\mu}_{0}(f)$ is continuous.*

PROOF. (i) Let $\mu$ and $\nu$ be probability distributions. Then we have

$$\begin{aligned}\left|\overline{\mathrm{D}}^{\mu}(\Pi)-\overline{\mathrm{D}}^{\nu}(\Pi)\right| &= \left|\sum_{x,y}\mu(x,y)|\Pi(x,y)|-\sum_{x,y}\nu(x,y)|\Pi(x,y)|\right|\\ &\leq \sum_{x,y}|\Pi(x,y)|\cdot|\mu(x,y)-\nu(x,y)|\\ &\leq \mathrm{D}(\Pi)\cdot||\mu-\nu||_1\ .\end{aligned}$$

(ii) Let $\epsilon>0$ be arbitrary. For every probability distribution $\sigma$ there exists a protocol $\Pi^{\sigma}_{\epsilon}$ such that $\mathrm{D}^{\sigma}_{0}(f)\geq\overline{\mathrm{D}}^{\sigma}(\Pi^{\sigma}_{\epsilon})-\epsilon/2$ and $\mathrm{D}(\Pi^{\sigma}_{\epsilon})<2^{|\mathcal{X}\times\mathcal{Y}|+1}=:D$. Define $\delta(\epsilon):=\epsilon/2(D+1)$. Then for arbitrary probability distributions $\mu$ and $\nu$ with $||\mu-\nu||_1<\delta(\epsilon)$ we have

$$\begin{aligned}\mathrm{D}^{\mu}_{0}(f)-\mathrm{D}^{\nu}_{0}(f) &\leq \overline{\mathrm{D}}^{\mu}(\Pi^{\nu}_{\epsilon})-\overline{\mathrm{D}}^{\nu}(\Pi^{\nu}_{\epsilon})+\epsilon/2\\ &\leq \mathrm{D}(\Pi^{\nu}_{\epsilon})\cdot||\mu-\nu||_1+\epsilon/2\\ &< \epsilon\ .\end{aligned}$$

Similarly, we obtain $\mathrm{D}^{\nu}_{0}(f)-\mathrm{D}^{\mu}_{0}(f)<\epsilon$. □

Lemmas 3.1.11 and 3.1.12 allow us to write

$$\max_{\mu\in\mathcal{M}}\mathrm{D}^{\mu}_{0}(f) \tag{3.1.13}$$

instead of

$$\sup_{\mu\in\mathcal{M}}\mathrm{D}^{\mu}_{0}(f)\ ,$$

because a continuous function takes on its extremal values on a compact set. Hence, we do not need sup-arguments in the sequel. The expression in (3.1.13) will turn out

to be equal to the average case randomized public coin communication complexity of $f$ defined in the next subsection. Clearly, for every $\epsilon > 0$ we can write

$$\max_{\mu \in \mathcal{M}} \mathrm{D}^{\mu}_{\epsilon}(f) \tag{3.1.14}$$

instead of

$$\sup_{\mu \in \mathcal{M}} \mathrm{D}^{\mu}_{\epsilon}(f) \ ,$$

because the set $\{\mathrm{D}^{\mu}_{\epsilon}(f) \mid \mu \in \mathcal{M}\}$ is bounded and only contains natural numbers. The expression (3.1.14) will turn out to be the worst case randomized public coin $\epsilon$-error communication complexity, again, defined in the next subsection.

We close this one with simple upper and lower bounds (see Kushilevitz & Nisan 1997, p. 6, Proposition 1.3 and Exercise 1.9):

Observation 3.1.15. *For every function $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ we have*

$$\lceil \log |\mathbf{range}(f)| \rceil \leq \mathrm{D}(f) \leq \min(\lceil \log |\mathcal{X}| \rceil , \lceil \log |\mathcal{Y}| \rceil) + \lceil \log |\mathbf{range}(f)| \rceil \ .$$

Of course, for Boolean functions the lower bound is useless. In general, much better bounds can be derived when we take on a combinatorial view on protocols in Section 3.2.

### 3.1.2 Randomized protocols

Often, randomized algorithms are both simpler and faster than every known deterministic algorithm solving the same problem. We refer the reader to Motwani & Raghavan (1995) for an excellent introduction to the exciting field of randomized algorithms. The same applies if one adds randomness to Yao's deterministic two player model. Randomized communication complexity was also defined in the seminal paper of Yao (1979). In the randomized model, the players are allowed to "toss coins" during the execution of a protocol, and the messages they send each other may also depend on the outcomes of the coin tosses. Consequently, the messages, the transcript and the computed function become random variables. We distinguish between two types of randomized protocols, namely "public coin" and "private coin" ones. In a public coin protocol, Alice and Bob share a common public coin whose outcomes are known to both players. In a private coin protocol, each player has its own random coin to flip. Important is that Alice cannot see Bob's coin flips and vice versa. While the latter model seems more realistic than the public coin model, it was shown by Newman (1991) that the models are essentially the same. We note that a randomized protocol can be interpreted as a probability distribution over deterministic protocols.

Definition 3.1.16 (Randomized protocol).

(i) *A* randomized public coin protocol $\Pi$ (over domain $\mathcal{X} \times \mathcal{Y}$ with range $\mathcal{Z}$) *is defined as a pair* $\Pi := (\Pi', \mathrm{C})$, *where* C *is a random variable over a finite set* $\mathcal{C}$, *and* $\Pi'$ *is a deterministic protocol over domain* $(\mathcal{X} \times \mathcal{C}) \times (\mathcal{Y} \times \mathcal{C})$ *with range* $\mathcal{Z}$. *The random variable* C *is called the* common coin.

(ii) *A* randomized private coin protocol $\Pi$ (over domain $\mathcal{X} \times \mathcal{Y}$ with range $\mathcal{Z}$) *is defined as a triple* $\Pi := (\Pi', \mathrm{A}, \mathrm{B})$, *where* A *and* B *are random variables over finite sets* $\mathcal{A}$ *and* $\mathcal{B}$, *respectively, and* $\Pi'$ *is a deterministic protocol over domain* $(\mathcal{X} \times \mathcal{A}) \times (\mathcal{Y} \times \mathcal{B})$ *with range* $\mathcal{Z}$. *The random variable* A *is called* Alice's coin, B *is called* Bob's coin.

Definition 3.1.17 (Transcript). *Let $\Pi$ be a randomized protocol. Given an input $(x, y)$, we define the* transcript $\Pi(x, y)$ *as the random variable* $\Pi(x, y) := \Pi'(x, \mathrm{C}, y, \mathrm{C})$, *if* $\Pi = (\Pi', \mathrm{C})$ *is a randomized public coin protocol, and as* $\Pi(x, y) := \Pi'(x, \mathrm{A}, y, \mathrm{B})$, *if* $\Pi = (\Pi', \mathrm{A}, \mathrm{B})$ *is a randomized private coin protocol.*

Definition 3.1.18 (Computed function). *Let $\Pi$ be a randomized protocol over domain $\mathcal{X} \times \mathcal{Y}$ with range $\mathcal{Z}$. We define the* function $f_\Pi$ computed by $\Pi$ *as the random variable* $f_\Pi := ((x, y) \mapsto f_{\Pi'}(x, \mathrm{C}, y, \mathrm{C}))$ *over* $\mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, *if* $\Pi = (\Pi', \mathrm{C})$ *is a randomized public coin protocol, and as* $f_\Pi := ((x, y) \mapsto f_{\Pi'}(x, \mathrm{A}, y, \mathrm{B}))$, *if* $\Pi = (\Pi', \mathrm{A}, \mathrm{B})$ *is a randomized private coin protocol.*

Definition 3.1.19 (Randomized communication cost). *Let $\Pi$ be a randomized protocol over domain $\mathcal{X} \times \mathcal{Y}$.*

(i) *The* worst case randomized communication cost of $\Pi$, $\mathrm{R}(\Pi)$, *is defined as*

$$\mathrm{R}(\Pi) := \max_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \max_{c\in\mathcal{C}} |\Pi'(x, c, y, c)| \ ,$$

*if* $\Pi = (\Pi', \mathrm{C})$ *is a randomized public coin protocol with common coin* C *defined over the finite set $\mathcal{C}$, and as*

$$\mathrm{R}(\Pi) := \max_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \max_{a\in\mathcal{A}, b\in\mathcal{B}} |\Pi'(x, a, y, b)| \ ,$$

*if* $\Pi = (\Pi', \mathrm{A}, \mathrm{B})$ *is a randomized private coin protocol with Alice's coin* A *defined over $\mathcal{A}$ and Bob's coin* B *defined over $\mathcal{B}$ for finite sets $\mathcal{A}$ and $\mathcal{B}$, respectively.*

(ii) *The* average case randomized communication cost of $\Pi$, $\overline{\mathrm{R}}(\Pi)$, *is defined as*

$$\overline{\mathrm{R}}(\Pi) := \max_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \mathrm{E}\left[\Pi(x, y)\right] \ ,$$

*where the expectation is over the coin tosses.*

Definition 3.1.20 ($\epsilon$-error).

(i) *Let $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ be a function, and let $\epsilon \geq 0$ be a real number. A randomized protocol $\Pi$ over domain $\mathcal{X} \times \mathcal{Y}$ with range $\mathcal{Z}$* computes *$f$* with *$\epsilon$*-error, *if for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ we have*

$$\Pr\left[f_\Pi(x, y) \neq f(x, y)\right] \leq \epsilon \ .$$

(ii) *Let $f\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{B}$ be a Boolean function, and let $\epsilon \geq 0$ be a real number. A randomized protocol $\Pi$ over domain $\mathcal{X} \times \mathcal{Y}$ with range $\mathbb{B}$* computes *$f$* with one-sided *$\epsilon$*-error, *if for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ we have*

$$\begin{aligned} f(x, y) = 0 \quad &\Longrightarrow \quad \Pr\left[f_\Pi(x, y) \neq f(x, y)\right] = 0 \ , \\ f(x, y) = 1 \quad &\Longrightarrow \quad \Pr\left[f_\Pi(x, y) \neq f(x, y)\right] \leq \epsilon \ . \end{aligned}$$

In the literature, a randomized protocol that computes a function with zero error is sometimes called a *Las Vegas protocol*, while a randomized protocol that computes a function with a bounded error is called *Monte Carlo protocol*.

Definition 3.1.21 (Randomized communication complexity).

(i) *Let $f$ be a function. The* average case randomized public coin communication complexity of $f$, $\mathrm{R}_0^{\mathrm{pub}}(f)$, *is defined as the infimum average case randomized communication cost of a randomized public coin protocol computing $f$ with* 0*-error, i. e.*

$$\mathrm{R}_0^{\mathrm{pub}}(f) := \inf\{\overline{\mathrm{R}}(\Pi) \mid \Pi \text{ a randomized public coin protocol computing } f \text{ with 0-error}\} .$$

(ii) *Let $f$ be a function, and let $\epsilon > 0$ be a real number. The* worst case randomized public coin $\epsilon$-error communication complexity of $f$, $\mathrm{R}_\epsilon^{\mathrm{pub}}(f)$, *is defined as the minimum worst case randomized communication cost of a randomized public coin protocol computing $f$ with $\epsilon$-error, i. e.*

$$\mathrm{R}_\epsilon^{\mathrm{pub}}(f) := \min\{\mathrm{R}(\Pi) \mid \Pi \text{ a randomized public coin protocol computing } f \text{ with } \epsilon\text{-error}\} .$$

(iii) *Let $f$ be a Boolean function, and let $\epsilon > 0$ be a real number. The* worst case randomized public coin one-sided $\epsilon$-error communication complexity of $f$, $\mathrm{R}_\epsilon^{\mathrm{pub},1}(f)$, *is defined as the minimum worst case randomized communication cost of a randomized public coin protocol computing $f$ with one-sided $\epsilon$-error, i. e.*

$$\mathrm{R}_\epsilon^{\mathrm{pub},1}(f) := \min\{\mathrm{R}(\Pi) \mid \Pi \text{ a randomized public coin protocol computing } f \text{ with one-sided } \epsilon\text{-error}\} .$$

For randomized private coin protocols one can define the complexity measures $\mathrm{R}_0(f) = \mathrm{R}_0^{\mathrm{priv}}(f)$, $\mathrm{R}_\epsilon(f) = \mathrm{R}_\epsilon^{\mathrm{priv}}(f)$ and $\mathrm{R}_\epsilon^1(f) = \mathrm{R}_\epsilon^{\mathrm{priv},1}(f)$ analogously to the ones for public coin protocols.

As mentioned above, public and private coin complexities are not far apart (see Kushilevitz & Nisan 1997, p. 33, Theorem 3.14; p. 34, Exercise 3.15):

Fact 3.1.22 (Newman). *Let $f\colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ be a Boolean function. For every $\epsilon > 0$ and $\delta > 0$ we have*

$$\begin{aligned} \mathrm{R}_{\epsilon+\delta}(f) &\leq \mathrm{R}_\epsilon^{\mathrm{pub}}(f) + \mathcal{O}\left(\log n + \log(1/\delta)\right) , \\ \mathrm{R}_0(f) &\leq \mathcal{O}\left(\mathrm{R}_0^{\mathrm{pub}}(f) + \log n\right) . \end{aligned}$$

Newman's result also shows that the size of the probability space can be restricted to $2^{\mathrm{polylog}(n)}$ for inputs of size $n$, if one allows a small increase in communication cost and error probability.

An important technique in the theory of randomized algorithms is *probability amplification*, i. e. one reduces the error probability of a randomized algorithm to an arbitrarily small constant by running the algorithm on the same input several times with independent coin tosses and then taking the majority vote of the outcomes. This can be done for randomized protocols, too.

The following fact can be found in (Köbler *et al.* 1993, p. 70, Lemma 2.14). We make use of this Chernoff-like result in Theorem 5.7.10.

Fact 3.1.23 (Probability amplification). *Let $E$ be an event that occurs with probability $\frac{1}{2} + \epsilon$, $0 < \epsilon \leq \frac{1}{2}$. Then $E$ occurs within $t$ independent trials ($t$ odd) at least $t/2$ times with probability at least*

$$1 - \frac{1}{2} \cdot (1 - 4 \cdot \epsilon^2)^{t/2} .$$

Many lower bound methods for randomized communication complexity are based on the following simple application of *Yao's Minimax-principle* (see e. g. Yao (1983) or Kushilevitz & Nisan 1997, p. 36, Theorem 3.20) relating randomized and distributional complexity:

FACT 3.1.24 (Yao). *For every Boolean function $f$ and every $\epsilon > 0$ we have*

$$\begin{aligned} \mathrm{R}_0^{\mathrm{pub}}(f) &= \max_\mu \mathrm{D}_0^\mu(f) \ , \\ \mathrm{R}_\epsilon^{\mathrm{pub}}(f) &= \max_\mu \mathrm{D}_\epsilon^\mu(f) \ . \end{aligned}$$

### 3.1.3 Counting protocols

Analogously to the Turing machine model, one can add the power of counting to Yao's model. The concept of counting means that the players can make nondeterministic guesses during a computation. Because on different guesses the output values may be different, one has to specify an *acceptance mode*, a predicate that tells us which inputs are considered to be accepted based on the number of accepting and rejecting computations.

There are several possibilities to define counting protocols (via proof systems, covers, etc.). We choose the following elegant variant due to the author:

DEFINITION 3.1.25 (Counting protocol). *A* counting protocol (over domain $\mathcal{X} \times \mathcal{Y}$) *is a deterministic protocol over domain $(\mathcal{X} \times \mathbb{B}^{g_A}) \times (\mathcal{Y} \times \mathbb{B}^{g_B})$ with range $\mathbb{B}$, where $g_A, g_B \geq 0$ are natural numbers denoting the lengths of the* guess strings.

Note that one could define counting protocols using abstract guess sets instead of $\mathbb{B}^{g_A}$ and $\mathbb{B}^{g_B}$, respectively. We do not do this, because the above definition corresponds more closely with the definition of complexity class operators in Section 5.2.

DEFINITION 3.1.26 (Communication cost). *The* worst case communication cost *of a counting protocol is defined as the worst case deterministic communication cost when viewed as a deterministic protocol.*

DEFINITION 3.1.27. *For a counting protocol $\Pi$ we denote with*

$$\begin{aligned} \mathrm{acc}_\Pi(x,y) &:= |\{\Pi((x,w_A),(y,w_B)) \mid f_\Pi((x,w_A),(y,w_B)) = 1\}| \ , \\ \mathrm{rej}_\Pi(x,y) &:= |\{\Pi((x,w_A),(y,w_B)) \mid f_\Pi((x,w_A),(y,w_B)) = 0\}| \ . \end{aligned}$$

*the* number of accepting (rejecting) transcripts of $\Pi$ on input $(x,y)$.

DEFINITION 3.1.28 (Computed function). *Given a counting protocol $\Pi$ and an acceptance mode $\mu$, the* function computed by $\Pi$ in $\mu$ acceptance mode, *$f_\Pi^\mu$, is defined as*

$$f_\Pi^\mu(x,y) := [\mu(\mathrm{acc}_\Pi(x,y), \mathrm{rej}_\Pi(x,y))] \ .$$

*We say that* $\Pi$ computes $f$ in $\mu$ acceptance mode, *if $f_\Pi^\mu = f$.*

We list the most prominent acceptance modes:

$$\begin{aligned} N^1(acc,rej) &:= (acc > 0) \ , \\ N^0(acc,rej) &:= (rej = 0) \ , \\ PP(acc,rej) &:= (acc > rej) \ , \\ \oplus P(acc,rej) &:= (acc \mod 2 = 1) \ . \end{aligned}$$

$N^1$ is the *nondeterministic*, $N^0$ the *co-nondeterministic*, $PP$ the *probabilistic* and $\oplus P$ the *parity acceptance mode.*

DEFINITION 3.1.29 (Counting complexities).

(i) *The* nondeterministic communication complexity of $f$, $\mathrm{N}^1(f)$, *is defined as the minimum worst case communication cost of a counting protocol computing $f$ in nondeterministic acceptance mode.*

(ii) *The* co-nondeterministic communication complexity of $f$, $\mathrm{N}^0(f)$, *is defined as the minimum worst case communication cost of a counting protocol computing $f$ in co-nondeterministic acceptance mode.*

(iii) *The* probabilistic communication complexity of $f$, $\mathrm{PP}(f)$, *is defined as the minimum worst case communication cost of a counting protocol computing $f$ in probabilistic acceptance mode.*

(iv) *The* parity communication complexity of $f$, $\oplus\mathrm{P}(f)$, *is defined as the minimum worst case communication cost of a counting protocol computing $f$ in parity acceptance mode.*

While the gap between counting complexities and worst case deterministic communication complexity can be exponential, it was shown in Aho *et al.* (1983) that the gap between $\mathrm{N}(f) := \max\{\mathrm{N}^0(f), \mathrm{N}^1(f)\}$ and $\mathrm{D}(f)$ is at most quadratic.

FACT 3.1.30 (Aho *et al.*). *For every Boolean function $f$ we have*

$$\mathrm{D}(f) = \mathcal{O}\left(\mathrm{N}^0(f)\mathrm{N}^1(f)\right) = \mathcal{O}\left(\mathrm{N}(f)^2\right) \ .$$

In addition, Las Vegas communication complexity $\mathrm{R}_0(f)$ is lower bounded by $\mathrm{N}(f)$.

FACT 3.1.31. *For every Boolean function $f$ we have*

$$\mathrm{N}(f) \leq \mathrm{R}_0(f) + 3 \ .$$

We give a proof of this fact for the reader's convenience, because none was provided in Kushilevitz & Nisan (1997).

PROOF. Let $\Pi := (\Pi', A, B)$ be an optimal randomized private coin protocol for $f$ with average case randomized communication cost $t := \mathrm{R}_0(f)$. We construct a nondeterministic one with cost at most $t + 3$: let Alice have $x$ and Bob $y$. Alice guesses a string $w$ of length $\leq t$ and sends its number to Bob using $t + 1$ many bits. They consider $w$ as a possible transcript of $\Pi$. Alice checks if there exists a random string $a$ such that $w$ is compatible with her part of the communication in $\Pi'$ on $(x, a)$. She tells Bob, if this is the case (1 bit). Bob does the same (1 bit). They accept the input $(x, y)$, if both have found random strings $a$ and $b$, respectively, such that $\Pi'((x, a), (y, b)) = w$ and $f_{\Pi'}((x, a), (y, b)) = 1$.

Thus, we have obtained

$$\mathrm{N}^1(f) \leq \mathrm{R}_0(f) + 3 \ .$$

Now, the fact follows from

$$\mathrm{N}^0(f) = \mathrm{N}^1(\overline{f}) \leq \mathrm{R}_0(\overline{f}) + 3 = \mathrm{R}_0(f) + 3 \ .$$

□

Both facts show that there is also an at most quadratic gap between $\mathrm{N}(f)$ and $\mathrm{R}_0(f)$, but we do not need this in the sequel.

Analogously to distributional deterministic communication complexity, we define a distributional parity communication complexity. The reason is that one can prove an analogous Minimax-statement (see Observation 3.1.40) for the BP-Parity-P complexity (see Definition 3.1.38) as for bounded error randomized communication complexity (see Fact 3.1.24).

Definition 3.1.32 (Computed function). *Let $f\colon \mathcal{X}\times\mathcal{Y}\to\mathbb{B}$ be a Boolean function, let $\mu$ be a probability distribution on $\mathcal{X}\times\mathcal{Y}$, and let $\epsilon>0$ be a real number. We say that a counting protocol* $\Pi$ computes $f$ in parity acceptance mode with $(\mu,\epsilon)$-distributional error, *if $f$ and $f_\Pi^{\oplus P}$ are $(\mu,\epsilon)$-close.*

Definition 3.1.33 (Distributional parity communication complexity).
*The* $(\mu,\epsilon)$-distributional parity communication complexity of $f$, $\oplus\mathrm{P}^\mu_\epsilon(f)$, *is defined as the minimum worst case communication cost of a counting protocol computing $f$ in parity acceptance mode with $(\mu,\epsilon)$-distributional error.*

Similar to Observation 3.1.10 we have

Observation 3.1.34. *Let $f$ be a Boolean function, and let $\epsilon>0$ be a real number. Then we have*

$$\oplus\mathrm{P}^\mu_\epsilon(f)=\min\{\oplus\mathrm{P}(\tilde{f})\mid \tilde{f} \text{ and } f \text{ are } (\mu,\epsilon)\text{-close}\}\ .$$

We will see later that interesting effects can occur when one combines counting with randomization.

Definition 3.1.35 (Randomized counting protocol). *A* randomized counting protocol $\Pi$ *(over domain $\mathcal{X}\times\mathcal{Y}$) is a probability distribution over counting protocols, i. e. $\Pi:=(\{\Pi_a\}_{a\in A},\alpha)$, where $\alpha$ is a random variable with values in $A$, and each $\Pi_a$, $a\in A$, is a counting protocol over domain $\mathcal{X}\times\mathcal{Y}$.*

Definition 3.1.36 (Communication cost). *The* worst case communication cost of a randomized counting protocol $\Pi:=(\{\Pi_a\}_{a\in A},\alpha)$ *is defined as the maximum worst case communication cost of the counting protocols $\Pi_a$ that have non-zero weight under the probability distribution on $A$ induced by $\alpha$.*

Definition 3.1.37 (Computed function). *Let $f\colon \mathcal{X}\times\mathcal{Y}\to\mathbb{B}$ be a Boolean function, and let $\epsilon>0$ be a real number. A randomized counting protocol* $\Pi$ computes $f$ in parity acceptance mode with $\epsilon$-error, *if for all $(x,y)\in\mathcal{X}\times\mathcal{Y}$ we have*

$$\Pr_\alpha\left[f^{\oplus\mathrm{P}}_{\Pi_\alpha}(x,y)\neq f(x,y)\right]\leq\epsilon\ .$$

*In this case we call* $\Pi$ a BP-Parity-P protocol computing $f$ with $\epsilon$-error.

Definition 3.1.38 (BP-Parity-P complexity). *Let $f$ be a Boolean function, and let $\epsilon>0$ be a real number. The* BP-Parity-P complexity of $f$, $\mathrm{BP}\oplus\mathrm{P}^{\mathrm{pub}}_\epsilon(f)$, *is defined as the minimum worst case communication cost of a BP-Parity-P protocol computing $f$ with $\epsilon$-error.*

Observation 3.1.39. *For every Boolean function $f$ and every real number $\epsilon>0$ the BP-Parity-P complexity can be upper-bounded by*

$$\mathrm{BP}\oplus\mathrm{P}^{\mathrm{pub}}_\epsilon(f)\leq\min\{\mathrm{D}(f),\oplus\mathrm{P}(f),\mathrm{R}^{\mathrm{pub}}_\epsilon(f)\}\ .$$

An adaptation of Fact 3.1.24 proves

Observation 3.1.40. *For every Boolean function $f$ and every real number $\epsilon>0$ we have*

$$\mathrm{BP}\oplus\mathrm{P}^{\mathrm{pub}}_\epsilon(f)=\max_\mu\oplus\mathrm{P}^\mu_\epsilon(f)\ .$$

### 3.1.4 Alternating protocols

The concept of alternation was originally defined for the Turing machine model as a generalization of nondeterminism. Alternation can be translated to Yao's model. This was done in (Babai *et al.* 1986, p. 339) by defining players East, West, North and South. We give an equivalent definition of alternating protocols.

In an alternating protocol the players may guess bits. Each state of the protocol is either *rejecting* (0), *accepting* (1), *existential* ($\exists$) or *universal* ($\forall$).

If a player guesses a bit in an existential state, then this guess is called existential; universal guesses are defined similarly.

Now, we are ready for a formal definition:

DEFINITION 3.1.41 (Alternating protocol). *An* alternating protocol (over domain $\mathcal{X} \times \mathcal{Y}$) *is a labeled binary tree, where leaves $v$ are labeled by $z_v \in \{0,1\}$ and inner nodes are labeled by $Q_v \in \{\exists, \forall\}$ and by functions $a_v \colon \mathcal{X} \to \{0,1,*\}$ or $b_v \colon \mathcal{Y} \to \{0,1,*\}$, respectively. Each inner node $v$ has two children $v_0$ and $v_1$.*

If in a run of an alternating protocol the players are in common state $v$ labeled by $a_v$, we say that Alice *guesses universally*, if $a_v(x) = *$ and $Q_v = \forall$, and that she *guesses existentially*, if $a_v(x) = *$ and $Q_v = \exists$. Analogously, if it is Bob's turn.

DEFINITION 3.1.42 (Alternating communication cost). *Let $\Pi$ be an alternating protocol. The* worst case alternating communication cost of $\Pi$, $\mathrm{A}(\Pi)$, *is defined as the maximum length of a path from the root to a leaf in the protocol tree of $\Pi$.*

DEFINITION 3.1.43 (Computed function). *Given an alternating protocol $\Pi$ over domain $\mathcal{X} \times \mathcal{Y}$, the* function computed by $\Pi$, $f_\Pi \colon \mathcal{X} \times \mathcal{Y} \to \mathbb{B}$, *is defined as follows: we associate with each node $v$ of $\Pi$ a function $f_v \colon \mathcal{X} \times \mathcal{Y} \to \mathbb{B}$ and define $f_\Pi$ as the function computed at the root. For a leaf $v$ we define $f_v(x,y) := z_v$. For an inner node $v$ labeled by $a_v$ we define*

$$f_v(x,y) := \begin{cases} f_{v_c}(x,y) & \text{, if } c := a_v(x) \in \{0,1\} \ , \\ [Q_v c \in \{0,1\} \colon f_{v_c}(x,y) = 1] & \text{, if } a_v(x) = * \ . \end{cases}$$

*Similarly, for inner nodes labeled by $b_v$.*

*An alternating protocol* $\Pi$ computes a function $f$, *if $f_\Pi = f$.*

We say that an alternating protocol has $k$ alternations if starting in an existential state the maximum number of alternations between existential and universal states on every path from the root to a leaf of the protocol tree is bounded by $k$.

DEFINITION 3.1.44 (Alternating communication complexity). *The* worst case alternating communication complexity of $f$, $\mathrm{A}(f)$, *is defined as the minimum worst case alternating communication cost of an alternating protocol computing $f$, i. e.*

$$\mathrm{A}(f) := \min\{\mathrm{A}(\Pi) \mid \Pi \text{ an alternating protocol computing } f\} \ .$$

With $\mathrm{A}^k(f)$ we denote the restriction to alternating protocols with $k$ alternations.

## 3.2 Covers and partitions

As mentioned earlier, the success in proving lower bounds for communication complexity measures stems from the combinatorial view we will take on protocols. The most fundamental notion in the combinatorics of protocols is that of the *(combinatorial) rectangle*:

Definition 3.2.1 (Combinatorial rectangle). *A* (combinatorial) rectangle *in $\mathcal{X} \times \mathcal{Y}$ is a subset $R \subseteq \mathcal{X} \times \mathcal{Y}$ such that $R = A \times B$ for some $A \subseteq \mathcal{X}$ and $B \subseteq \mathcal{Y}$.*

*Given a rectangle $R = C \times D$ we define $A(R) := C$ and $B(R) := D$, respectively.*

We fix a deterministic protocol $\Pi$. Recall that $\mathrm{L}_\Pi$ denotes the set of leaves of the protocol tree of $\Pi$, and that $R_v$ denotes the set of inputs that reach the node $v$.

First of all, $\Pi$ partitions the input space via $\{R_l \mid l \in \mathrm{L}_\Pi\}$, because every input reaches exactly one leaf. Secondly, every $R_v$ is a rectangle, because at the root we start with the rectangle $\mathcal{X} \times \mathcal{Y}$ and at an inner node $v$ either $A(R_v)$ is changed, if Alice speaks, or $B(R_v)$, if Bob speaks, respectively. Thus, we have obtained

Fact 3.2.2. *A deterministic protocol partitions the input space into a set of rectangles.*

Now, if $\Pi$ computes a function $f$, then $f$ has to be constant on each rectangle at a leaf of $\Pi$. We call such values colors.

Definition 3.2.3 (Color). *Let $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ be a function, let $R \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ be a relation, let $S$ be a subset of $\mathcal{X} \times \mathcal{Y}$, and let $z \in \mathcal{Z}$ be a* color.

(i) *$S$ is called* $z$-chromatic (with respect to $f$), *if $f(S) = \{z\}$.*

(ii) *$S$ is called* $z$-chromatic (with respect to $R$), *if for each legal input $(x, y) \in S$ we have $(x, y, z) \in R$.*

(iii) *$S$ is called* monochromatic *if it is $z$-chromatic for a color $z \in \mathcal{Z}$.*

We define several combinatorial measures that lower bound deterministic and also (co-)nondeterministic communication complexities.

Definition 3.2.4 (Cover numbers). *Let $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ be a function, and let $z \in \mathcal{Z}$ be a color.*

(i) *The* protocol partition number of $f$, $\mathrm{C}^{\mathrm{P}}(f)$, *is defined as the smallest number of leaves in a protocol computing $f$.*

(ii) *The* partition number of $f$, $\mathrm{C}^{\mathrm{D}}(f)$, *is defined as the smallest number of monochromatic rectangles with respect to $f$ in a partition of $\mathcal{X} \times \mathcal{Y}$.*

(iii) *The* $z$-chromatic partition number of $f$, $\mathrm{C}^{\mathrm{D},z}(f)$, *is the smallest number of $z$-chromatic rectangles with respect to $f$ in a partition of the $z$-inputs of $f$.*

(iv) *The* cover number of $f$, $\mathrm{C}(f)$, *is the smallest number of monochromatic rectangles with respect to $f$ needed to cover $\mathcal{X} \times \mathcal{Y}$ (possibly with intersections).*

(v) *The* $z$-chromatic cover number of $f$, $\mathrm{C}^z(f)$, *is the smallest number of $z$-chromatic rectangles with respect to $f$ needed to cover the $z$-inputs of $f$.*

We immediately get

Fact 3.2.5. *Let $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ be a function. Then we have*

(i) $\mathrm{C}(f) \leq \mathrm{C}^{\mathrm{D}}(f) \leq \mathrm{C}^{\mathrm{P}}(f) \leq 2^{\mathrm{D}(f)}$.

(ii) $\mathrm{C}(f) = \sum_{z \in \mathcal{Z}} \mathrm{C}^z(f)$, $\mathrm{C}^{\mathrm{D}}(f) = \sum_{z \in \mathcal{Z}} \mathrm{C}^{\mathrm{D},z}(f)$.

The following folklore fact is obtained via balancing a protocol with few leaves. This yields a new protocol with communication cost not far away from the logarithm of the number of leaves of the old protocol.

Fact 3.2.6. *For every function* $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ *we have*

$$\log \mathrm{C}^{\mathrm{P}}(f) \leq \mathrm{D}(f) \leq 3 \log \mathrm{C}^{\mathrm{P}}(f) \ .$$

For a proof see (Kushilevitz & Nisan 1997, p. 19, Lemma 2.9; p. 20, Exercise 2.9).

A long-standing open problem in communication complexity is, whether a similar relation holds for the partition number, the so-called $\mathrm{C}^{\mathrm{D}}$-vs.-$\mathrm{C}^{\mathrm{P}}$-problem (see Kushilevitz & Nisan 1997, p. 20, Open Problem 2.10).

Open Question 3.2.7 ($\mathrm{C}^{\mathrm{D}}$-vs.-$\mathrm{C}^{\mathrm{P}}$-problem). *Is the quantity* $\log \mathrm{C}^{\mathrm{D}}(f)$ *linearly related to* $\log \mathrm{C}^{\mathrm{P}}(f)$ *?*

*In particular, this would imply* $\mathrm{D}(f) = \mathcal{O}\left(\log \mathrm{C}^{\mathrm{D}}(f)\right)$.

For (co-)nondeterministic communication complexity we obtain a result similar to Fact 3.2.6, this time via $z$-chromatic cover numbers.

Fact 3.2.8. *For every Boolean function* $f\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{B}$ *and every* $z \in \{0,1\}$ *we have*

$$\log \mathrm{C}^{z}(f) \leq \mathrm{N}^{z}(f) \leq \log \mathrm{C}^{z}(f) + 2 \ .$$

Again, we give a proof of this fact, because none was provided in Kushilevitz & Nisan (1997).

Proof. A counting protocol can be considered as a family of deterministic protocols. Each induces a partition of the input space. Thus, the union of the $z$-chromatic rectangles is a $z$-chromatic cover. This gives the lower bound. For the upper bound, Alice guesses the number of a rectangle $R$ in an optimal cover and sends this number to Bob. Then, Alice sends $[x \in A(R)]$ and Bob sends $[y \in B(R)]$. They accept if both sent a one. □

## 3.3 Lower bounds

The complexity measures introduced in the preceding sections are hard to calculate, because in general it is extremely expensive to enumerate all protocols computing a function in order to find one with minimal communication cost. This is why for each complexity measure $M$ one tries to find combinatorial measures $M' \leq M$ that are easily computable and (hopefully) close to $M$. The cover numbers introduced in the last section are an intermediate step in this direction.

### 3.3.1 Rectangle size method

For worst case (co-)nondeterministic communication complexity such a combinatorial measure is the *rectangle size method*, and, as a special case, the *fooling set method*. For definitions, applications and proofs we refer the reader to (Kushilevitz & Nisan 1997, Sections 1.3 and 2.4).

Definition 3.3.1 (Rectangle size method). *Let* $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ *be a function, let* $z \in \mathcal{Z}$ *be a color, and let* $\mu$ *be a probability distribution on the* $z$*-inputs* $f^{-1}(z)$.

$$\begin{aligned} \mathrm{mono}^{\mu,z}(f) &:= \max\{\mu(R) \mid R \text{ a } z\text{-chromatic rectangle with respect to } f\} \ , \\ \mathrm{mono}^{z}(f) &:= \min\{\mathrm{mono}^{\mu,z}(f) \mid \mu \text{ a probability distribution on } f^{-1}(z)\} \ , \end{aligned}$$

$$\mathrm{rsm}^{z}(f) := \log\left(\frac{1}{\mathrm{mono}^{z}(f)}\right) \ .$$

*Let $\mu$ be a probability distribution on the input space $\mathcal{X} \times \mathcal{Y}$.*

$$\begin{aligned} \operatorname{mono}^{\mu}(f) &:= \max\{\mu(R) \mid R \text{ a monochromatic rectangle with respect to } f\} \ , \\ \operatorname{mono}(f) &:= \min\{\operatorname{mono}^{\mu}(f) \mid \mu \text{ a probability distribution on } \mathcal{X} \times \mathcal{Y}\} \ , \end{aligned}$$

$$\operatorname{rsm}(f) := \log\left(\frac{1}{\operatorname{mono}(f)}\right) \ .$$

FACT 3.3.2. *For every Boolean function $f\colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ and every $z \in \{0,1\}$ we have*

$$\frac{1}{\operatorname{mono}^z(f)} \leq \mathrm{C}^z(f) \leq \mathcal{O}\left(\frac{n}{\operatorname{mono}^z(f)}\right) \ ,$$

*and*

$$\frac{1}{\operatorname{mono}(f)} \leq \mathrm{C}(f) \leq \mathcal{O}\left(\frac{n}{\operatorname{mono}(f)}\right) \ .$$

Combining Fact 3.2.8 and Fact 3.3.2 we obtain

FACT 3.3.3 (Characterization). *For every Boolean function $f\colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ and every $z \in \{0,1\}$ we have*

$$\operatorname{rsm}^z(f) \leq \mathrm{N}^z(f) \leq \operatorname{rsm}^z(f) + \mathcal{O}(\log n) \ ,$$

*and*

$$\operatorname{rsm}(f) \leq \mathrm{N}(f) \leq \operatorname{rsm}(f) + \mathcal{O}(\log n) \ .$$

Because this method even characterizes worst case (co-)nondeterministic communication complexity up to an additive logarithmic term, worst case (co-)nondeterministic communication complexity is well understood. One can show (see Fact 3.1.30) that the rectangle size method also leads to lower bounds for worst case deterministic communication complexity that are at most a quadratic factor lower than the true value.

### 3.3.2 Lower bound methods for randomized communication complexity

Many lower bound methods have been developed for randomized communication complexity. The most prominent ones are the *discrepancy method* (Kushilevitz & Nisan 1997, p. 38, Section 3.5), the *Fourier method* of Raz (1995), the *$\epsilon$-monochromatic rectangle size method* (or *corruption method*, see e. g. Beame *et al.* 2006), and the *factorization norm method* of Linial & Shraibman (2007). The latter paper showed that most known lower bounds for randomized communication complexity are actually lower bounds for quantum communication complexity. Klauck (2001) gave a characterization of the PP complexity via the discrepancy method. Another approach to lower bounds for randomized communication complexity are information-theoretical methods. We discuss them in the next chapter.

### 3.3.3 Rank method

The most important method for worst case deterministic communication complexity, the *rank method*, was introduced in Mehlhorn & Schmidt (1982). The basic idea is to consider a function $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ as a matrix $M^f$ and rectangles as submatrices of $M^f$. Consider a partition of $\mathcal{X} \times \mathcal{Y}$ into monochromatic rectangles $R_1, \ldots, R_t$. Because the rectangles form a partition of the input space, the matrix $M^f$ can be written as a sum $\sum z_i \cdot M_i$ of $t$ rank one matrices $M_1, \ldots, M_t$, where $(M_i)_{x,y} := [(x,y) \in R_i]$, i. e.

$M_i$ has value one on the rectangle inputs and zero otherwise. Thus, $\mathbb{F}$-rank$(M^f) \leq t$ for every suitable field $\mathbb{F}$ with $\mathcal{Z} \subseteq \mathbb{F}$.

For a relation $R$, we can also associate with $R$ a matrix $M^R$. But here, the rank method does not work.

Definition 3.3.4 (Communication matrices).

(i) *We associate with every function* $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ *a matrix* $M^f$ *of dimensions* $|\mathcal{X}| \times |\mathcal{Y}|$. *The rows of* $M^f$ *are indexed by the elements of* $\mathcal{X}$ *and the columns are indexed by the elements of* $\mathcal{Y}$. *The* $(x, y)$*-entry* $M^f_{x,y}$ *of* $M^f$ *is simply defined as* $f(x, y)$.

(ii) *Given a relation* $R \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, *we associate with* $R$ *a matrix* $M^R\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{P}(\mathcal{Z})$ *defined as*

$$M^R_{x,y} := \{z \in \mathcal{Z} \mid (x, y, z) \in R\} \ .$$

The considerations above yield

Fact 3.3.5 (Mehlhorn & Schmidt). *For every function* $f$ *we have*

$$\log \mathbb{R}\text{-rank}(M^f) \leq \mathrm{D}(f) \ .$$

It is still open whether the rank method is polynomially tight for Boolean functions. (For arbitrary functions one can show exponential gaps.)

Open Question 3.3.6 (Logarithmic rank conjecture). *Do we have*

$$\mathrm{D}(f) \leq (\log \mathbb{R}\text{-rank}(M^f))^{\mathcal{O}(1)}$$

*for every Boolean function* $f$*?*

Non-constant gaps have been shown in Nisan & Wigderson (1995); Raz & Spieker (1993).

The logarithmic rank conjecture for modular communication complexity was proved in (Damm *et al.* 2004, Proposition 5.3). Using a different characterization via intersection graphs the same result had been established earlier implicitly by Pudlák & Rödl (1994).

Fact 3.3.7. *Let* $f$ *be a Boolean function. Then we have*

$$\log \mathbb{F}_2\text{-rank}(M^f) \leq \oplus\mathrm{P}(f) \leq \log \mathbb{F}_2\text{-rank}(M^f) + \mathcal{O}(1) \ .$$

We will use this important fact twice in Chapter 5: first of all, in the derivation of a lower bound method for the BP-Parity-P complexity, and secondly, when we consider the Parity-P complexity of problems based on quasi-random graph families.

# 4 Information complexity

## 4.1 Introduction

An important question in communication complexity is, how to prove lower bounds for average case deterministic communication complexity. One very successful approach that has led to striking results is the study of *information complexity measures*.

It is natural to ask for protocols that on average reveal as little information as possible about the inputs while computing $f$ (or something close to $f$).

For randomized protocols $\Pi$ this leads to the notion of *average case information cost of* $\Pi$, $\mathrm{IC}^{\mu}(\Pi)$, i. e. the mutual information between the inputs and the transcript of the protocol on those inputs, and, for a function $f$, to the *average case randomized $\epsilon$-error information complexity of $f$*, $\mathrm{IC}_{\epsilon}^{\mathrm{rand},\mu}(f)$, as the infimum average case information cost of a private coin randomized protocol computing $f$ with $\epsilon$-error.

Note that it does not make sense to define this notion for public coin protocols, because in this case the players can use the outcomes of the public coin as a one-time pad. Thus, for every function $f$, every distribution $\mu$ and every $\epsilon \geq$ we have

$$\mathrm{IC}_{\epsilon}^{\mathrm{rand,pub},\mu}(f) = 0 \ ,$$

not a very impressive lower bound.

Average case randomized information complexity was first explicitly defined in the work of Chakrabarti *et al.* (2001), where a direct sum theorem was shown in the simultaneous message passing model. In Harsha *et al.* (2007); Jain *et al.* (2003) this was extended to direct sum theorems for $k$-round randomized communication complexity in Yao's model. We also mention that notions of randomized information complexity have led to an optimal quadratic separation of $\mathrm{R}_0(f)$ and $\mathrm{N}(f)$ (see Jayram *et al.* 2003) and an elegant alternative proof of the celebrated theorem of Kalyanasundaram & Schnitger (1992) that $\mathrm{R}_{\epsilon}(\mathrm{DISJ}_n)$ is linear in $n$. See Bar-Yossef *et al.* (2004).

The restriction to deterministic protocols leads to the notion of *average case deterministic information complexity of $f$*, $\mathrm{IC}_0^{\mathrm{det},\mu}(f)$.

For deterministic protocols we are not sure whom to give credit for introducing information-theoretical methods to lower bound average case deterministic communication complexity. We could trace such ideas back to Orlitsky & El Gamal (1990), where (in our terminology) it was shown that $\mathrm{IC}_0^{\mathrm{det},\mu}(f)$ is a lower bound for $\mathrm{D}_0^{\mu}(f)$. Ahlswede & Cai (1994) showed (again, in our terminology)

$$\mathrm{D}_0^{\mu}(f) \leq \mathrm{IC}_0^{\mathrm{det},\mu}(f) + \inf\{\mathrm{d}(P) \mid P \text{ protocol induced partition } \}$$

as an upper bound, where the *depth* $\mathrm{d}(P)$ is based on partition refinements.

In the next section we show that average case deterministic information complexity and average case deterministic communication complexity actually differ only by a constant factor:

$$\mathrm{D}_0^{\mu}(f) = \Theta\left(\mathrm{IC}_0^{\mathrm{det},\mu}(f)\right) \ . \tag{4.1.1}$$

Such a result is by no means obvious, since it is not clear how from a protocol with low information cost one should obtain one with small average case complexity. We obtain this result via a balancing technique inspired by the proof of Fact 3.2.6.

Equation (4.1.1) can be interpreted as a non-trivial generalization of Shannon's *Noiseless Coding Theorem* telling us that in order to communicate a random variable

$X$ the average code length is between the Shannon entropy $\mathrm{H}(X)$ and $\mathrm{H}(X)+1$. We said non-trivial, because applying Shannon's Theorem to each message sent by the players only yields

$$\overline{\mathrm{D}}^{\mu}(\Pi) \leq \mathrm{IC}^{\mu}(\Pi) + k$$

for $k$-round protocols. In contrast, our result holds for protocols with arbitrarily many rounds.

It is an open problem, whether a result like (4.1.1) holds for average case randomized information complexity defined over unbounded round protocols.

## 4.2 Average case deterministic information complexity

Let $\mathrm{H}(U)$ denote the *Shannon entropy* of a random variable $U$, and let $\mathrm{I}(U;V) = \mathrm{H}(V) - \mathrm{H}(V \mid U)$ denote the *Shannon mutual information* between the random variables $U$ and $V$. If $\mu$ is a probability distribution on a finite probability space $\Omega$, then $\mathrm{H}(\mu) := \sum_{\omega\in\Omega} \mu(\omega) \cdot \log(1/\mu(\omega))$. We write $U \sim \mu$, if the random variable is distributed according to the probability distribution $\mu$. For an introduction to information theory, we refer the reader to the excellent monograph of Cover & Thomas (1991).

Definition 4.2.1 (Average case information cost). *Let $\Pi$ be a protocol, and let $\mu$ be a probability distribution on the input space $\mathcal{X}\times\mathcal{Y}$. Then the* $\mu$-average case information cost of $\Pi$, $\mathrm{IC}^{\mu}(\Pi)$, *is defined as the Shannon mutual information between transcript and input, i. e.*

$$\mathrm{IC}^{\mu}(\Pi) := \mathrm{I}((X,Y);\Pi(X,Y)) \ ,$$

*where $(X,Y)\sim\mu$.*

Since $\Pi(X,Y)$ is a function of $(X,Y)$ for deterministic protocols $\Pi$, we have

$$\begin{aligned}
\mathrm{IC}^{\mu}(\Pi) &= \mathrm{I}((X,Y);\Pi(X,Y)) \\
&= \mathrm{H}(\Pi(X,Y)) - \mathrm{H}(\Pi(X,Y) \mid (X,Y)) \\
&= \mathrm{H}(\Pi(X,Y)) - 0 \\
&= \mathrm{H}(\Pi(X,Y)) \ .
\end{aligned}$$

That means, the information cost of $\Pi$ is just the entropy of the distribution which weighs the leaves of $\Pi$ with the weights of the associated rectangles according to the probability distribution $\mu$.

Definition 4.2.2 (Average case deterministic information complexity).
*Let $f\colon \mathcal{X}\times\mathcal{Y}\to\mathcal{Z}$ be a function, and let $\mu$ be a probability distribution on the input space $\mathcal{X}\times\mathcal{Y}$. The* $\mu$-average case deterministic information complexity of $f$, $\mathrm{IC}_0^{\mathrm{det},\mu}(f)$, *is defined as the infimum $\mu$-average case information cost of a deterministic protocol computing $f$, i. e.*

$$\mathrm{IC}_0^{\mathrm{det},\mu}(f) := \inf\{\mathrm{IC}^{\mu}(\Pi) \mid \Pi \text{ a deterministic protocol computing } f\} \ .$$

Definition 4.2.3 (Average code length). *Let $\mathcal{C}\colon \mathcal{M}\to\{0,1\}^*$ be a code of $\mathcal{M}$, i. e. an injective mapping, and let $\mu$ be a probability distribution on $\mathcal{M}$. The* $\mu$-average code length of $\mathcal{C}$, $\mathrm{L}^{\mu}(\mathcal{C})$, *is defined as*

$$\mathrm{L}^{\mu}(\mathcal{C}) := \mathrm{E}_M\left(|\mathcal{C}(M)|\right) \ ,$$

*where $M\sim\mu$.*

We only consider *prefix codes*, i. e. codes $\mathcal{C}$ such that if $u,v$ are different elements of $\mathcal{C}(M)$ then $u$ is not a prefix of $v$. It is well-known that the optimal average code length for $M$ is essentially the entropy of $\mu$.

Definition 4.2.4 (Minimal average prefix code length). *Let $\mathcal{M}$ be a set, and let $\mu$ be a probability distribution on $\mathcal{M}$. The* minimal $\mu$-average prefix code length of $\mathcal{M}$, $\overline{\mathrm{L}}^{\mu}(\mathcal{M})$, *is defined as*

$$\overline{\mathrm{L}}^{\mu}(\mathcal{M}) := \inf\{\mathrm{L}^{\mu}(\mathcal{C}) \mid \mathcal{C} \text{ a prefix code of } \mathcal{M}\} \ .$$

Fact 4.2.5 (Shannon). *Let $\mu$ be a probability distribution on $\mathcal{M}$. Then we have*

$$\mathrm{H}(\mu) \leq \overline{\mathrm{L}}^{\mu}(\mathcal{M}) \leq \mathrm{H}(\mu) + 1 \ .$$

Let $\Pi$ be a deterministic protocol. Recall that $\mathrm{P}_{\Pi}$ denotes the partition of $\mathcal{X} \times \mathcal{Y}$ into rectangles induced by $\Pi$, $\mathrm{V}_{\Pi}$ is the set of the nodes of the protocol tree of $\Pi$, $\mathrm{L}_{\Pi}$ is the set of leaves of the protocol tree of $\Pi$, and $\mathrm{R}_v$ is the set of inputs going through the node $v \in \mathrm{V}_{\Pi}$. If $p\colon \mathrm{L}_{\Pi} \to [0,1]$ is a probability distribution on the leaves of $\Pi$, define $\tilde{p}(v)$ as the sum of the weights of the leaves according to $p$ in the subtree with root $v$.

The following theorem can be found in Orlitsky & El Gamal (1990). We state and prove it here for the sake of completeness:

Theorem 4.2.6. *Let $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ be a function, and let $\mu$ be a probability distribution on the input space $\mathcal{X} \times \mathcal{Y}$. Then we have*

$$\mathrm{IC}_0^{\mathrm{det},\mu}(f) \leq \mathrm{D}_0^{\mu}(f) \ .$$

Proof. For every $\epsilon > 0$ there exists a deterministic protocol $\Pi$ computing $f$ such that $\mathrm{D}_0^{\mu}(f) \geq \overline{\mathrm{D}}^{\mu}(\Pi) - \epsilon$. Let $\mathcal{C}\colon \mathrm{P}_{\Pi} \to \{0,1\}^*$ be the assignment of transcripts of $\Pi$ to the corresponding combinatorial rectangles. Then $\mathcal{C}$ is a prefix code. For the partition $\mathrm{P}_{\Pi}$ of $\mathcal{X} \times \mathcal{Y}$ define the probability distribution $\nu\colon \mathrm{P}_{\Pi} \to [0,1]$ as $\nu(R) := \mu(R)$ for all $R \in \mathrm{P}_{\Pi}$. Using Shannon's Theorem (Fact 4.2.5) one obtains

$$\overline{\mathrm{D}}^{\mu}(\Pi) = \mathrm{L}^{\nu}(\mathcal{C}) \geq \overline{\mathrm{L}}^{\nu}(\mathrm{P}_{\Pi}) \geq \mathrm{H}(\nu) = \mathrm{H}(\Pi(X,Y)) = \mathrm{IC}^{\mu}(\Pi) \geq \mathrm{IC}_0^{\mathrm{det},\mu}(f) \ ,$$

where $(X,Y) \sim \mu$. As $\epsilon$ can be made arbitrarily small, the theorem follows. □

Definition 4.2.7. *For $p = (p_1, \ldots, p_l)$ with $0 \leq p_1, \ldots, p_l \leq 1$ and $\sum_{i\in[l]} p_i \leq 1$ define*

$$\mathrm{H}(p) := \textstyle\sum_{i\in[l]} p_i \cdot \log(1/p_i) + \left(1 - \sum_{i\in[l]} p_i\right) \cdot \log\left(1/\left(1 - \sum_{i\in[l]} p_i\right)\right) \ .$$

*In case $\sum_{i\in[l]} p_i = 1$ this reduces to the standard definition of Shannon entropy for the probability distribution $(p_1, \ldots, p_l)$.*

The following Lemma 4.2.8 is needed below. Although it is a special case of the well-known chain rule ($\mathrm{H}(X,Y) = \mathrm{H}(X) + \mathrm{H}(Y \mid X)$), we also state and prove it here for the sake of completeness:

Lemma 4.2.8. *Let $(p_1, \ldots, p_l)$ be a probability weight vector (i. e. $\sum_{i\in[l]} p_i = 1$), and let $I \subseteq [l]$. Define $q := \sum_{i\in I} p_i$. Then*

$$\mathrm{H}\left((p_i)_{i\in[l]}\right) = \mathrm{H}\left((p_i)_{i\in I}\right) + (1-q) \cdot \mathrm{H}\left((p_j/1-q)_{j\in[l]-I}\right) \ , \tag{4.2.9}$$

$$\mathrm{H}\left((p_i)_{i\in I}\right) = \mathrm{H}(1-q) + q \cdot \mathrm{H}\left((p_i/q)_{i\in I}\right) \ , \tag{4.2.10}$$

$$\mathrm{H}\left((p_i)_{i\in[l]}\right) = \mathrm{H}(q) + q \cdot \mathrm{H}\left((p_i/q)_{i\in I}\right) + (1-q) \cdot \mathrm{H}\left((p_j/1-q)_{j\in[l]-I}\right) \ . \tag{4.2.11}$$

PROOF. Equation (4.2.9) follows from

$$\begin{aligned}
\mathrm{H}\left((p_i)_{i\in[l]}\right) &= \sum_{i\in[l]} p_i \cdot \log\left(\frac{1}{p_i}\right) \\
&= \sum_{i\in I} p_i \cdot \log\left(\frac{1}{p_i}\right) + (1-q)\cdot\log\left(\frac{1}{1-q}\right) \\
&\quad + \sum_{i\in[l]-I} p_i\cdot\log(1-q) + \sum_{i\in[l]-I} p_i\cdot\log\left(\frac{1}{p_i}\right) \\
&= \mathrm{H}\left((p_i)_{i\in I}\right) + (1-q)\cdot\mathrm{H}\left((\frac{p_j}{1-q})_{j\in[l]-I}\right) \ .
\end{aligned}$$

Equation (4.2.10) is a special case of (4.2.9). Equation (4.2.11) follows from (4.2.9), (4.2.10) and $\mathrm{H}(q) = \mathrm{H}(1-q)$. □

From information theory we know that the entropy of a weight distribution on the leaves of a (protocol) tree is a lower bound for the minimal average (protocol) depth. For codes an optimal upper bound can be realized using Huffman trees. There, the step from a distribution to a code is almost trivial. In contrast, the tight relationship between the minimal average protocol depth (average case deterministic communication complexity) and the minimal entropy of a protocol induced weight distribution (average case deterministic information complexity) shown below is far from obvious. Of course, the Huffman construction does not work here. Instead, we present a balancing technique analogous to the one used to establish Fact 3.2.6.

THEOREM 4.2.12. *Let $f\colon \mathcal{X}\times\mathcal{Y}\to\mathcal{Z}$ be a function, and let $\mu$ be a probability distribution on the input space $\mathcal{X}\times\mathcal{Y}$. Then we have*

$$\mathrm{D}_0^{\mu}(f) \le \frac{2}{\mathrm{H}(1/3)}\cdot \mathrm{IC}_0^{\mathrm{det},\mu}(f) + 3 \ .$$

PROOF. Let $\Pi$ be a deterministic protocol computing $f$. From $\Pi$ we build another protocol $\mathcal{A}_\Pi(p)$, which is more balanced than $\Pi$ according to the probability distribution $p\colon \mathrm{L}_\Pi \to [0,1]$: Each player knows $\Pi$, $p$ and his part of the input. The players proceed as follows, distinguishing between three cases:

*Case 1*: There exists a leaf $l\in\mathrm{L}_\Pi$ such that $p(l)=1$. Each player sends one bit telling if his part of the input is *compatible with* $\mathrm{R}_l = A_l\times B_l$, i.e. $x\in A_l$, $y\in B_l$. If this is the case, the players take the value $z_l$ of $l$ in $\Pi$ as the function value and stop the protocol. (2 bits of communication.) Otherwise, they know that they have a zero-weight input. They start $\Pi$ to determine the function value. (No average case cost.)

*Case 2*: There exists a leaf $l\in\mathrm{L}_\Pi$ such that $\frac{2}{3} < p(l) < 1$. Again, each player sends one bit telling if his part of the input is compatible with $\mathrm{R}_l = A_l\times B_l$. If this is the case, the players take the color $z_l$ of $l$ in $\Pi$ as the function value and stop the protocol. (2 bits of communication.)
Otherwise the players recursively proceed with $\mathcal{A}_\Pi(p_{|\neg l})$ on their input, where for $v\in\mathrm{V}_\Pi$ the probability distribution $p_{|\neg v}\colon \mathrm{L}_\Pi\to[0,1]$ is defined as

$$p_{|\neg v}(l) := \begin{cases} 0 & \text{, if } l \text{ is a leaf of the subtree of } \Pi \text{ induced by } v, \\ \frac{p(l)}{1-\tilde{p}(v)} & \text{, otherwise.} \end{cases}$$

(As stated above, $\tilde{p}(v)$ abbreviates the sum of the weights of the leaves below $v$.)

*Case 3*: For all $l\in\mathrm{L}_\Pi\colon p(l)\le\frac{2}{3}$. The players choose a node $v\in\mathrm{V}_\Pi$ that satisfies $\frac{1}{3}\le\tilde{p}(v)\le\frac{2}{3}$. Such a node exists, since all leaves have weight less than $\frac{2}{3}$. (Beginning

at the root they follow a path choosing in each step the son with bigger weight until they reach a node $v$ with $\frac{1}{3} \leq \tilde{p}(v) \leq \frac{2}{3}$.) Each player sends one bit telling if his part of the input is compatible with $\mathrm{R}_v$. If this is the case, the players recursively proceed with $\mathcal{A}_\Pi(p_{|v})$ on their input, where $p_{|v} \colon \mathrm{L}_\Pi \to [0,1]$ is defined as

$$p_{|v}(l) := \begin{cases} \frac{p(l)}{\tilde{p}(v)} & \text{, if } l \text{ is a leaf of the subtree of } \Pi \text{ induced by } v, \\ 0 & \text{, otherwise.} \end{cases}$$

Otherwise, i. e. if $(x,y) \notin \mathrm{R}_v$, the players recursively proceed with $\mathcal{A}_\Pi(p_{|\neg v})$ on their input.

Let $p \colon \mathrm{L}_\Pi \to [0,1]$ be a probability distribution on $\mathrm{L}_\Pi$, and let $\mu_p$ be the corresponding probability distribution on the inputs, i. e. $\mu_p(x,y) := \frac{p(l)}{|\mathrm{R}_l|}$ if $(x,y) \in \mathrm{R}_l$ . Denote by $\overline{\mathrm{D}}(p)$ the $\mu_p$-average case deterministic communication cost of $\mathcal{A}_\Pi(p)$. The following recurrence is immediate from the construction.

1. If there exists a leaf $l \in \mathrm{L}_\Pi$ such that $p(l) = 1$, then $\overline{\mathrm{D}}(p) = 2$.
2. If there exists a leaf $l \in \mathrm{L}_\Pi$ such that $\frac{2}{3} < p(l) < 1$, then
$$\overline{\mathrm{D}}(p) = 2 + (1 - p(l)) \cdot \overline{\mathrm{D}}(p_{|\neg l}) \ .$$
3. If for all $l \in \mathrm{L}_\Pi \colon p(l) \leq \frac{2}{3}$, then
$$\overline{\mathrm{D}}(p) = 2 + \tilde{p}(v) \cdot \overline{\mathrm{D}}(p_{|v}) + (1 - \tilde{p}(v)) \cdot \overline{\mathrm{D}}(p_{|\neg v}) \ .$$

We prove by induction on $\mathbf{nz}(p) := |\{l \in \mathrm{L}(\Pi) \mid p(l) > 0\}|$ that

$$\overline{\mathrm{D}}(p) \leq \frac{2}{\mathrm{H}(1/3)} \cdot \mathrm{H}(p) + 3 \ . \tag{4.2.13}$$

The *base case* $\mathbf{nz}(p) = 1$ is trivial, as in *Case 1* one has $\overline{\mathrm{D}}(p) = 2$. In the *induction step* $\mathbf{nz}(p) > 1$ *Cases 2* and *3* can occur:

*Case 2*:

$$\begin{aligned} \overline{\mathrm{D}}(p) &= 2 + (1 - p(l)) \cdot \overline{\mathrm{D}}(p_{|\neg l}) \\ &\overset{\text{(I.H.)}}{\leq} \frac{2}{\mathrm{H}(1/3)} \cdot (1 - p(l)) \cdot \mathrm{H}(p_{|\neg l}) + 2 + (1 - p(l)) \cdot 3 \\ &\leq \frac{2}{\mathrm{H}(1/3)} \cdot (1 - p(l)) \cdot \mathrm{H}(p_{|\neg l}) + 3 \\ &\leq \frac{2}{\mathrm{H}(1/3)} \cdot \big((1 - p(l)) \cdot \mathrm{H}(p_{|\neg l}) + \mathrm{H}(p(l))\big) + 3 \\ &\overset{\text{Lemma 4.2.8(2)}}{=} \frac{2}{\mathrm{H}(1/3)} \cdot \mathrm{H}(p) + 3 \ . \end{aligned}$$

*Case 3*: Observe that $\mathrm{H}(1/3) \leq \mathrm{H}(\tilde{p}(v))$ for $\frac{1}{3} \leq \tilde{p}(v) \leq \frac{2}{3}$. This entails

$$\begin{aligned} \overline{\mathrm{D}}(p) &= 2 + \tilde{p}(v) \cdot \overline{\mathrm{D}}(p_{|v}) + (1 - \tilde{p}(v)) \cdot \overline{\mathrm{D}}(p_{|\neg v}) \\ &\overset{\text{(I.H.)}}{\leq} \frac{2}{\mathrm{H}(1/3)} \cdot \big(\tilde{p}(v) \cdot \mathrm{H}(p_{|v}) + (1 - \tilde{p}(v)) \cdot \mathrm{H}(p_{|\neg v}) + \mathrm{H}(1/3)\big) \\ &\quad + \tilde{p}(v) \cdot 3 + (1 - \tilde{p}(v)) \cdot 3 \\ &\leq \frac{2}{\mathrm{H}(1/3)} \cdot \big(\tilde{p}(v) \cdot \mathrm{H}(p_{|v}) + (1 - \tilde{p}(v)) \cdot \mathrm{H}(p_{|\neg v}) + \mathrm{H}(\tilde{p}(v))\big) + 3 \\ &\overset{\text{Lemma 4.2.8(3)}}{=} \frac{2}{\mathrm{H}(1/3)} \cdot \mathrm{H}(p) + 3 \ . \end{aligned}$$

For every $\epsilon > 0$ there exists a deterministic protocol $\Pi$ computing $f$ such that $\mathrm{IC}_0^{\mathrm{det},\mu}(f) = \mathrm{IC}^\mu(\Pi) - \epsilon$. We define $\nu\colon \mathrm{L}_\Pi \to [0,1]$ as $\nu(l) := \mu(\mathrm{R}_l)$. Note that $\mathcal{A}_\Pi(\nu)$ is a deterministic protocol that computes $f$, and that $\mathrm{L}_\Pi = \mathrm{L}_{\mathcal{A}_\Pi(\nu)}$. Thus,

$$\begin{aligned}
\mathrm{D}_0^\mu(f) &\leq \overline{\mathrm{D}}^\mu(\mathcal{A}_\Pi(\nu)) \\
&= \overline{\mathrm{D}}^{\mu_\nu}(\mathcal{A}_\Pi(\nu)) \\
&= \overline{\mathrm{D}}(\nu) \\
&\overset{(4.2.13)}{\leq} \frac{2}{\mathrm{H}(1/3)} \cdot \mathrm{H}(\nu) + 3 \\
&= \frac{2}{\mathrm{H}(1/3)} \cdot \mathrm{IC}^\mu(\Pi) + 3 \\
&\leq \frac{2}{\mathrm{H}(1/3)} \cdot \left(\mathrm{IC}_0^{\mathrm{det},\mu}(f) + \epsilon\right) + 3 \ .
\end{aligned}$$

As $\epsilon$ can be made arbitrarily small, the theorem follows. □

Note that the result can be easily extended to the $k$-player NIH and NOF models of communication complexity, see (Kushilevitz & Nisan 1997, Chapter 6). Instead of the multiplicative factor 2, one gets a factor of $k$.

## 4.3 Lower bounds for public coin Las Vegas communication complexity

In this section we use the information-theoretical lower bound derived in Theorem 4.2.6 to show improved lower bounds for $\mathrm{R}_0^{\mathrm{pub}}(f)$ by the rectangle size method as defined in Section 3.3.

Combining the results in Fact 3.3.3, Fact 3.1.31 and Fact 3.1.22 one obtains

Fact 4.3.1. *Let $f\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{B}$ be a Boolean function with $|\mathcal{X}|, |\mathcal{Y}| \leq 2^n$. Then there exists a constant $C \geq 1$ such that*

$$\mathrm{R}_0^{\mathrm{pub}}(f) = \Omega\left(\mathrm{rsm}(f) - C \cdot \log n\right) \ .$$

Proof. We have

$$\mathrm{R}_0(f) = \mathcal{O}\left(\mathrm{R}_0^{\mathrm{pub}}(f) + \log n\right)$$

by Fact 3.1.22. Thus, there exists a constant $C$ such that

$$\begin{aligned}
\mathrm{R}_0^{\mathrm{pub}}(f) &= \Omega\left(\mathrm{R}_0(f) - C \cdot \log n\right) \\
&= \Omega\left(\mathrm{N}(f) - C \cdot \log n\right) \\
&= \Omega\left(\mathrm{rsm}(f) - C \cdot \log n\right) \ .
\end{aligned}$$

□

Lemma 4.3.2. *For every function $f$ we have*

$$\mathrm{D}_0^\mu(f) \geq \log\left(\frac{1}{\mathrm{mono}^\mu(f)}\right) \ .$$

PROOF. Given an arbitrary function $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, let $\Pi$, $\epsilon$ and $\nu$ be defined as in the proof of Theorem 4.2.6. Then, by Theorem 4.2.6, we have $\mathrm{D}_0^\mu(f) \geq \mathrm{H}(\nu) - \epsilon$ and

$$\begin{aligned} \mathrm{H}(\nu) &= \sum_{R \in \mathrm{P}_\Pi} \nu(R) \cdot \log\left(\frac{1}{\nu(R)}\right) \\ &\geq \sum_{R \in \mathrm{P}_\Pi} \nu(R) \cdot \log\left(\frac{1}{\mathrm{mono}^\mu(f)}\right) \\ &= \log\left(\frac{1}{\mathrm{mono}^\mu(f)}\right) \ . \end{aligned}$$

Again, as $\epsilon$ can be made arbitrarily small, the lemma follows. □

The lower bound for $\mathrm{R}_0^{\mathrm{pub}}$ obtained in Fact 4.3.1 can be improved by a constant factor:

THEOREM 4.3.3. *For every function $f$ we have*

$$\mathrm{R}_0^{\mathrm{pub}}(f) \geq \mathrm{rsm}(f) \ .$$

PROOF. By Yao's Minimax-principle (Fact 3.1.24) one has

$$\begin{aligned} \mathrm{R}_0^{\mathrm{pub}}(f) &= \max_\mu \mathrm{D}_0^\mu(f) \\ &\overset{\text{Lemma 4.3.2}}{\geq} \max_\mu \log\left(\frac{1}{\mathrm{mono}^\mu(f)}\right) \\ &= \mathrm{rsm}(f) \ . \end{aligned}$$

□

Using the improved bound (Theorem 4.3.3) we now have

COROLLARY 4.3.4. $\mathrm{R}_0^{\mathrm{pub}}(f) \geq \mathrm{N}(f) - \log n - \mathcal{O}(1)$.

For some functions this yields sharp (nonimprovable) bounds, which could not be obtained by Fact 4.3.1 alone. In particular, all rectangle size bounds and all fooling set bounds immediately apply to $\mathrm{R}_0^{\mathrm{pub}}$, no constants involved.

EXAMPLE 4.3.5. Let the *maximum function* $\mathrm{MAX}_n$, the *greater than function*, $\mathrm{GT}_n$, the *equality function*, $\mathrm{EQ}_n$, the *disjointness function*, $\mathrm{DISJ}_n$, and the *inner product function mod 2*, $\mathrm{IP}_n$, be defined as in Exercises 1.5 and 1.22 and Examples 1.21, 1.23 and 1.25, respectively, in Kushilevitz & Nisan (1997). (The latter will also be discussed in Section 5.4.) Then $\mathrm{R}_0^{\mathrm{pub}}(\mathrm{MAX}_n) \geq \log n$ and $\mathrm{R}_0^{\mathrm{pub}}(\mathrm{GT}_n), \mathrm{R}_0^{\mathrm{pub}}(\mathrm{EQ}_n), \mathrm{R}_0^{\mathrm{pub}}(\mathrm{DISJ}_n) > n$ and $\mathrm{R}_0^{\mathrm{pub}}(\mathrm{IP}_n) \geq n - 1$. In Buhrman *et al.* (2000) a lower bound of $n - \mathcal{O}(1)$ was obtained for $\mathrm{IP}_n$ via the incompressibility method. ◊

## 4.4 Concluding remarks

In this chapter we have seen that average case deterministic information complexity and average case deterministic communication complexity only differ by a constant factor. Does a similar relation hold for average case randomized information complexity, i. e.

OPEN QUESTION 4.4.1. *Do we have*

$$\mathrm{D}_0^\mu(f) = \Theta\left(\mathrm{IC}_0^{\mathrm{rand},\mu}(f)\right) \ ?$$

# 5 Structural communication complexity

## 5.1 Introduction

The field of *structural complexity theory* is so broad and rich that we do not make any attempt to give an overview of this field or at least to list the most important results. As an excuse, we would like to cite (Hemaspaandra & Ogihara 2002, p. 263), where they say that

> "it would be impossible to define or collect the field's most important theorems "

in their appendix (A Rogues' Gallery of Complexity Classes) that has a size of 40 pages! Instead, for introductions to (parts of) structural complexity we refer the reader to the excellent monographs of Balcázar *et al.* (1990, 1995); Du & Ko (2000); Köbler *et al.* (1993); Schöning (1986). Good surveys on a variety of topics in this field can be found in Selman (1988); Selman & Hemaspaandra (1997), especially on counting complexity in Schöning (1988) and Fortnow (1997), respectively.

To a complexity theorist, structure is meaning. In order to understand computational resources and their relationships one groups families of problems into *complexity classes* that can be solved with a certain computational power stemming from the resources one has added to the model under consideration.

Classically, in structural complexity theory one considers the Turing machine model and models of Boolean and algebraic circuits. If one adds the resources randomization, counting or alternation to the Turing machine model, one obtains standard complexity classes like *deterministic polynomial time*, **P**, *nondeterministic polynomial time*, **NP**, *co-nondeterministic polynomial time*, **coNP**, *bounded error probabilistic polynomial time*, **BPP**, *unbounded error probabilistic polynomial time*, **PP**, *parity polynomial time* $\oplus\mathbf{P}$, (*Parity-P* for short), *the polynomial-time hierarchy*, $\mathbf{PH} = \bigcup_{k\in\mathbb{N}} \Sigma_k^p$, and *polynomial space*, **PSPACE**. By their very definition one obtains a set of standard inclusions (see Table 5.1.1).

| | | | | | | |
|---|---|---|---|---|---|---|
| **P** | $\subseteq$ | **BPP** | $\subseteq$ | **PP** | $\subseteq$ | **PSPACE** , |
| **P** | $\subseteq$ | **NP**, **coNP** | $\subseteq$ | **PH** | $\subseteq$ | **PSPACE** , |
| **P** | $\subseteq$ | **PP**, $\oplus$**P** | $\subseteq$ | **PSPACE** . | | |

Table 5.1.1: Standard inclusions

Many complexity classes are (or can be formulated as) *counting classes*. These classes are based on Turing machines that can guess bits together with a fixed *acceptance mode* $\mu$. Let $\mathrm{acc}_T(x)$ and $\mathrm{rej}_T(x)$ denote the number of accepting and rejecting computations of $T$ on input $x$, respectively. Then an input is accepted by $T$ in $\mu$ acceptance mode, if $\mu(\mathrm{acc}_T(x), \mathrm{rej}_T(x))$ is true. A prominent example of such a counting class is $\oplus\mathbf{P}$, defined by Papadimitriou & Zachos (1983), where it was shown that $\oplus\mathbf{P}(\oplus\mathbf{P}) = \oplus\mathbf{P}$. Here, the acceptance mode gives "true", if the number of accepting computations is odd.

The classes **PH** and **PSPACE** can be defined via the concept of alternation: problems in **PH** are decidable with a constant number of alternations, problems in **PSPACE** with an efficient number (polynomial in the input size). An alternating Turing machine can guess bits universally and existentially. An input is accepted, if all successor configurations of a universal guess are accepting, and if for every existential guess there exists an accepting successor configuration.

One can define operators on complexity classes, e. g. the useful BP-operator, which was defined by Schöning (1989). Using the BP-operator and a relativized version of the so-called *Valiant-Vazirani-Lemma* (see Valiant & Vazirani 1986) Toda (1991) was able to prove his celebrated theorems establishing the inclusions

$$\mathbf{PH} \subseteq \mathrm{BP} \cdot \oplus\mathbf{P} \subseteq \mathbf{P}(\#\mathbf{P}) = \mathbf{P}(\mathbf{PP}) \ .$$

They tell us that counting (mod 2) plus the use of a random source is at least as powerful as the whole polynomial-time hierarchy **PH**, and that the same applies to the closure of **PP** under polynomial time Turing reductions. See also Schöning (1991) for a proof sketch diaphanously presenting the main ideas.

| | | | | | | |
|---|---|---|---|---|---|---|
| $\mathbf{P}$ | $\subsetneq$ (?) | $\mathbf{BPP}$ | $\subsetneq$ (?) | $\mathbf{PP}$ | $\subsetneq$ (?) | $\mathbf{PSPACE}$ , |
| $\mathbf{P}$ | $\subsetneq$ (?) | $\mathbf{NP}, \mathbf{coNP}$ | $\subsetneq$ (?) | $\mathbf{PH}$ | $\subsetneq$ (?) | $\mathbf{PSPACE}$ , |
| $\mathbf{P}$ | $\subsetneq$ (?) | $\mathbf{PP}, \oplus\mathbf{P}$ | $\subsetneq$ (?) | $\mathbf{PSPACE}$ , | | |

and what about the pairs **NP** vs. **coNP**(?), **NP** vs. **PP**(?), **NP** vs. $\oplus$**P**(?), or **PP** vs. $\oplus$**P**(?), ...

Table 5.1.2: Unknown inclusion relationships

Research in *structural communication complexity* started with the work of Babai *et al.* (1986), where some analogies between the Turing machine classes mentioned above and corresponding communication complexity classes $\mathbf{P}^{cc}$, $\mathbf{NP}^{cc}$, $\mathbf{PP}^{cc}$, $\oplus\mathbf{P}^{cc}$, $\mathbf{PSPACE}^{cc}$, $\mathbf{PH}^{cc} = \bigcup_{k\in\mathbb{N}} \Sigma_k^{cc}$, etc. were shown. Interestingly, while (almost) nothing is known about the standard classes in the Turing machine model (see Table 5.1.2), almost everything is known about the inclusion relationships between the respective communication complexity classes (see Table 5.1.3). One of the few exceptions is the long-standing open problem, whether the polynomial hierarchy is strictly contained in polynomial space or not.

| | | | | | | |
|---|---|---|---|---|---|---|
| $\mathbf{P}^{cc}$ | $\subsetneq$ | $\mathbf{BPP}^{cc}$ | $\subsetneq$ | $\mathbf{PP}^{cc}$ | $\subsetneq$ | $\mathbf{PSPACE}^{cc}$ , |
| $\mathbf{P}^{cc}$ | $\subsetneq$ | $\mathbf{NP}^{cc}, \mathbf{coNP}^{cc}$ | $\subsetneq$ | $\mathbf{PH}^{cc}$ , | | |
| $\mathbf{P}^{cc}$ | $\subsetneq$ | $\mathbf{PP}^{cc}, \oplus\mathbf{P}^{cc}$ | $\subsetneq$ | $\mathbf{PSPACE}^{cc}$ . | | |

The following pairs are incomparable:
($\mathbf{NP}^{cc}, \mathbf{co\text{-}NP}^{cc}$), ($\mathbf{NP}^{cc}, \mathbf{PP}^{cc}$), ($\mathbf{NP}^{cc}, \oplus\mathbf{P}^{cc}$), ($\mathbf{PP}^{cc}, \oplus\mathbf{P}^{cc}$).

Table 5.1.3: Known inclusion relationships

For more ground work, especially on closure properties, the boolean communication hierarchy, or counting communication complexity classes like $\mathrm{MOD}_m\mathbf{P}^{cc}$, see Halstenberg & Reischuk (1990) or Damm *et al.* (2004). Klauck (2003) established separation results between the classes $\mathbf{MA}^{cc}$ and $\mathbf{NP}^{cc}$, $\mathbf{MA}^{cc}$ and $\mathbf{APP}^{cc}$, and $\mathbf{APP}^{cc}$ and $\mathbf{PP}^{cc}$, respectively. In recent research, Buhrman *et al.* (2007) showed $\Sigma_2^{cc}, \Pi_2^{cc} \not\subseteq \mathbf{PP}^{cc}$. This was improved to $\Sigma_2^{cc}, \Pi_2^{cc} \not\subseteq \mathbf{UPP}^{cc}$ by Razborov & Sherstov (2008).

Because of the distributive nature of communication complexity, formal languages are defined a bit differently than usual. The *set of pairs of strings of equal length* is denoted by $\mathbb{B}^{**} := \{(x,y) \mid x,y \in \mathbb{B}^*, |x| = |y|\}$. A *language* $L$ is a subset of $\mathbb{B}^{**}$, its *$n$-bit section* $L_n$ is the set of all pairs $(x,y) \in L$ of $n$-bit words $x,y$.

A *(communication) complexity class* is a set of languages. Because our bounds on communication will use floors, ceilings and logarithms, the set of polynomials is not expressive enough, and we have to define $\mathbf{poly} := \{f\colon \mathbb{R}^+ \to \mathbb{R}^+ \mid \exists \text{ polynomial } p\colon f \leq p\}$, the *set of functions with polynomial growth*.

A family of functions $f := (f_n)_{n\in\mathbb{N}}$, $f_n\colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$, can be considered as a family of characteristic functions that defines a language $L_f := \{(x,y) \in \mathbb{B}^{**} \mid f_{|x|}(x,y) = 1\}$. For the other direction, a language $L$ defines a family of functions $\chi^L := (\chi^{L_n})_{n\in\mathbb{N}}$, where $\chi^{L_n}\colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$, $\chi^{L_n}(x,y) := [(x,y) \in L_n]$.

In the sequel, we often do not distinguish between languages and characteristic function families. Especially, for a complexity measure $M$ we write $M(L_n)$, where it should correctly read $M(\chi^{L_n})$.

We call a protocol over domain $\mathbb{B}^n \times \mathbb{B}^n$ an *$n$-bit* protocol. A protocol family $(\Pi_n)_{n\in\mathbb{N}}$ of $n$-bit protocols $\Pi_n$ *decides* a language $L$ if each $\Pi_n$ computes the characteristic function of $L_n$.

In each structural theory, the standard set of complexity classes is defined based on the standard set of complexity measures (deterministic, randomized, nondeterministic, etc.) and a notion of *efficiency*. In structural communication complexity, if a problem can be solved with communication complexity polylogarithmically in the input size, then we consider this as efficient.

Definition 5.1.1 (Some standard classes).

$$\begin{aligned}
\mathbf{P}^{cc} &:= \{L \mid \exists p \in \mathbf{poly}\colon \mathrm{D}(L_n) \leq p(\log n)\} &,\\
\mathbf{BPP}^{cc} &:= \{L \mid \exists p \in \mathbf{poly}\colon \mathrm{R}^{\mathrm{pub}}_{1/3}(L_n) \leq p(\log n)\} &,\\
\mathbf{PP}^{cc} &:= \{L \mid \exists p \in \mathbf{poly}\colon \mathrm{PP}(L_n) \leq p(\log n)\} &,\\
\mathbf{NP}^{cc} &:= \{L \mid \exists p \in \mathbf{poly}\colon \mathrm{N}^1(L_n) \leq p(\log n)\} &,\\
\mathbf{coNP}^{cc} &:= \{L \mid \exists p \in \mathbf{poly}\colon \mathrm{N}^0(L_n) \leq p(\log n)\} &,\\
\oplus\mathbf{P}^{cc} &:= \{L \mid \exists p \in \mathbf{poly}\colon \oplus\mathrm{P}(L_n) \leq p(\log n)\} &.
\end{aligned}$$

In the Turing machine model the complexity classes **PSPACE** and **PH** are defined based on the resource "space". The important observation that these classes can be defined via alternating Turing machines opened the possibility to define analogous classes in structural communication complexity.

Definition 5.1.2 (Alternating classes).

$$\begin{aligned}
\mathbf{PSPACE}^{cc} &:= \{L \mid \exists p \in \mathbf{poly}\colon \mathrm{A}(L_n) \leq p(\log n)\} &,\\
\mathbf{PH}^{cc} &:= \bigcup_{k\geq 0} \Sigma_k^{cc} &,\\
\Sigma_0^{cc} &:= \mathbf{P}^{cc} &,\\
\Sigma_{k+1}^{cc} &:= \{L \mid \exists p \in \mathbf{poly}\colon \mathrm{A}^k(L_n) \leq p(\log n)\} &, k \geq 0 \ .
\end{aligned}$$

From the plethora of function classes we only need the class *Sharp-P*, $\#\mathbf{P}^{cc}$, in the sequel. It contains all function families $\mathrm{acc}_\Pi := (\mathrm{acc}_{\Pi_n})_{n\in\mathbb{N}}$ defined by protocol families $\Pi := (\Pi_n)_{n\in\mathbb{N}}$ of $n$-bit counting protocols $\Pi_n$ that are efficient, i. e. there exists a $p \in \mathbf{poly}$ such that for all $n$ the communication cost of $\Pi_n$ is bounded by $p(\log n)$.

An important concept in structural complexity is *relativization*. Analogously to oracle Turing machines one can define *oracle protocols*. A deterministic, randomized,

counting or alternating protocol $\Pi$ over $\mathcal{X} \times \mathcal{Y}$ is an *oracle protocol* with oracle family $O = (O_m)_{m\in\mathbb{N}}$, if $\Pi$ contains *oracle nodes* in its protocol tree. Associated with an oracle node $v$ are two functions $a_v\colon \mathcal{X} \to \mathbb{B}^{m_v}$ and $b_v\colon \mathcal{Y} \to \mathbb{B}^{m_v}$. If Alice and Bob reach an oracle node $v$ during a computation on input $(x, y) \in X \times \mathcal{Y}$, they compute by themselves $x' := a_v(x)$ and $y' := b_v(y)$, respectively, and call $O_{m_v}$ on $(x', y')$. The oracle node $v$ has exactly $|\mathbf{range}(O_{m_v})|$ many successors. Alice and Bob continue the computation on one of them according to the returned value $O_{m_v}(x', y')$. The communication cost for each oracle call is $\lceil \log |\mathbf{range}(O_{m_v})| \rceil$. *Relativized communication complexity classes* are defined via efficient oracle protocol families. For example, $\mathbf{P}^{cc}(L')$ contains all languages $L$ which can be decided by an efficient protocol family $(\Pi_n)_{n\in\mathbb{N}}$ of deterministic $n$-bit oracle protocols $\Pi_n$ with oracle family $(L'_m)_{m\in\mathbb{N}}$.

Reductions play a central role in structural complexity. In Babai *et al.* (1986) different kinds of reductions were defined analogously to the Turing machine model. In structural communication complexity, many-one reductions defined below are also called *rectangular reductions*.

Definition 5.1.3 (Reductions). *Let $L$ and $L'$ be languages.*

*(i) $L$ is* many-one reducible *to $L'$, if there exist a bound $b \in$ **poly** and a family of function pairs $\{(f_n, g_n)\}_{n\in\mathbb{N}}$, $f_n, g_n\colon \mathbb{B}^n \to \mathbb{B}^{\lceil 2^{b(\log n)}\rceil}$, such that for all $n$-bit input pairs $(x, y)$ we have*

$$(x, y) \in L \iff (f_n(x), g_n(y)) \in L' \ .$$

*(ii) $L$ is* Turing reducible *to $L'$, if $L \in \mathbf{P}^{cc}(L')$.*

*(iii) $L$ is* majority reducible *to $L'$, if there exist bounds $b, t \in$ **poly** and a family of function pairs $\{(f_n, g_n)\}_{n\in\mathbb{N}}$, $f_n, g_n\colon \mathbb{B}^n \to \mathbb{B}^*$, such that for all $n$-bit input pairs $(x, y)$ we have*

$$\begin{aligned} f_n(x) &= \langle x_1, \ldots, x_t\rangle \ , \\ g_n(y) &= \langle y_1, \ldots, y_t\rangle \ , \end{aligned}$$

*where $t \le \lceil t(\log n)\rceil$, $|x_i| = |y_i| \le \lceil 2^{b(\log n)}\rceil$ and*

$$(x, y) \in L \iff (x_i, y_i) \in L' \text{ for the majority of the indices } i \in [\lceil t(\log n)\rceil].$$

*(iv) $L$ is* conjunctively reducible *to $L'$, if there exist bounds $b, t \in$ **poly** and a family of function pairs $\{(f_n, g_n)\}_{n\in\mathbb{N}}$, $f_n, g_n\colon \mathbb{B}^n \to \mathbb{B}^*$, such that for all $n$-bit input pairs $(x, y)$ we have*

$$\begin{aligned} f_n(x) &= \langle x_1, \ldots, x_t\rangle \ , \\ g_n(y) &= \langle y_1, \ldots, y_t\rangle \ , \end{aligned}$$

*where $t \le \lceil t(\log n)\rceil$, $|x_i| = |y_i| \le \lceil 2^{b(\log n)}\rceil$ and*

$$(x, y) \in L \iff (x_i, y_i) \in L' \text{ for all indices } i \in [\lceil t(\log n)\rceil].$$

## 5.2 Complexity class operators

In Section 5.3 we state an operator-theoretical version of the Lemma of Valiant & Vazirani (1986) that enables us to prove Toda's Theorems in Section 5.5. In order to formulate and prove the respective statements we must first define several complexity class operators and state some of their properties. For readers familiar with such

operators and their properties this might be a dry topic and they might want to skip this section.

A careful reader familiar with randomized communication complexity might wonder why the operators below are defined in a *public coin style*, i. e. both players get the same witness/random string. Of course, one can define the operators such that each player gets his/her own witness/random string (*private coin style*). The reason is that these definitions are equivalent, if the operators are simulated by a protocol. Alice can guess Bob's witness and send it to him, or she can send him her random string, because the length of witnesses/random strings is bounded polylogarithmically in the length of the input.

Definition 5.2.1 (Complexity class operators). *For a language $L$ and a bound $p \in \mathbf{poly}$ we define*

$$\begin{aligned}
\forall^p(L) &:= \{(x,y) \in \mathbb{B}^{**} \mid \forall w \in \mathbb{B}^{\lceil p(\log|x|)\rceil} \colon (\langle x,w\rangle, \langle y,w\rangle) \in L\} \ , \\
\exists^p(L) &:= \{(x,y) \in \mathbb{B}^{**} \mid \exists w \in \mathbb{B}^{\lceil p(\log|x|)\rceil} \colon (\langle x,w\rangle, \langle y,w\rangle) \in L\} \ , \\
\mathrm{Mod}_k^p(L) &:= \{(x,y) \in \mathbb{B}^{**} \mid |\{w \in \mathbb{B}^{\lceil p(\log|x|)\rceil} \mid (\langle x,w\rangle, \langle y,w\rangle) \in L\}| \mod k \neq 0\}, \\
\oplus^p(L) &:= \mathrm{Mod}_2^p(L) \ .
\end{aligned}$$

*For a communication complexity class $\mathcal{C}$ we define*

$$\begin{aligned}
\mathrm{co}\cdot\mathcal{C} &:= \{\overline{L} \mid L \in \mathcal{C}\} \ , \\
\forall\cdot\mathcal{C} &:= \{\forall^p(L) \mid L \in \mathcal{C}, p \in \mathbf{poly}\} \ , \\
\exists\cdot\mathcal{C} &:= \{\exists^p(L) \mid L \in \mathcal{C}, p \in \mathbf{poly}\} \ , \\
\mathrm{Mod}_k\cdot\mathcal{C} &:= \{\mathrm{Mod}_k^p(L) \mid L \in \mathcal{C}, p \in \mathbf{poly}\} \ , \\
\oplus\cdot\mathcal{C} &:= \mathrm{Mod}_2\cdot\mathcal{C} \ .
\end{aligned}$$

*We also define the communication complexity version of the BP-operator introduced in Schöning (1989):*

*A language $L$ is in $\mathrm{BP}\cdot\mathcal{C}$ if there exist a language $L' \in \mathcal{C}$ and a bound $q \in \mathbf{poly}$ such that for all $n$-bit input pairs $(x,y)$ we have*

$$\begin{aligned}
(x,y) \in L &\implies |\{r \in \mathbb{B}^{\lceil q(\log n)\rceil} \mid (\langle x,r\rangle, \langle y,r\rangle) \in L'\}|/2^{\lceil q(\log n)\rceil} \geq 2/3 \ , \\
(x,y) \notin L &\implies |\{r \in \mathbb{B}^{\lceil q(\log n)\rceil} \mid (\langle x,r\rangle, \langle y,r\rangle) \in L'\}|/2^{\lceil q(\log n)\rceil} \leq 1/3 \ .
\end{aligned}$$

The following observation shows that the names used for the operators are compatible with the names of classical communication complexity classes, if the operators are applied to $\mathbf{P}^{cc}$.

Observation 5.2.2 (Compatibility).

$$\begin{aligned}
\mathbf{NP}^{cc} &= \exists\cdot\mathbf{P}^{cc} \ , & \mathrm{Mod}_k\mathbf{P}^{cc} &= \mathrm{Mod}_k\cdot\mathbf{P}^{cc} \ , \\
\mathbf{coNP}^{cc} &= \forall\cdot\mathbf{P}^{cc} \ , & \oplus\mathbf{P}^{cc} &= \oplus\cdot\mathbf{P}^{cc} \ , \\
\mathbf{BPP}^{cc} &= \mathrm{BP}\cdot\mathbf{P}^{cc} \ .
\end{aligned}$$

Observation 5.2.3. $\mathrm{BP}\cdot\oplus\mathbf{P}^{cc} = \{L \mid \exists p \in \mathbf{poly} \colon \mathrm{BP}{\oplus}\mathrm{P}^{\mathrm{pub}}_{1/3}(L_n) \leq p(\log n)\}$.

We observe the following properties of the communication complexity class operators. The respective proofs are so easy that we omit most of them for brevity.

Observation 5.2.4 (Probability amplification). *Let $\mathcal{C}$ be a communication complexity class closed under majority reductions, and let $b \in \mathbf{poly}$. If a language $L$ is in $\mathrm{BP}\cdot\mathcal{C}$, then there exist a language $L' \in \mathcal{C}$ and a bound $q \in \mathbf{poly}$ such that for all $n$-bit input pairs $(x, y)$ we have*

$$(x,y) \in L \implies |\{r \in \mathbb{B}^{\lceil q(\log n)\rceil} \mid (\langle x, r\rangle, \langle y, r\rangle) \in L'\}| \,/\, 2^{\lceil q(\log n)\rceil} \geq 1 - 2^{-b(\log n)},$$
$$(x,y) \notin L \implies |\{r \in \mathbb{B}^{\lceil q(\log n)\rceil} \mid (\langle x, r\rangle, \langle y, r\rangle) \in L'\}| \,/\, 2^{\lceil q(\log n)\rceil} \leq 2^{-b(\log n)} \ .$$

Observation 5.2.5 (Inclusion). *Let $\mathcal{C}$ be a communication complexity class that is closed under many-one reductions. Then for every operator $\mathrm{Op} \in \{\forall, \exists, \mathrm{Mod}_k, \oplus, \mathrm{BP}\}$ we have $\mathcal{C} \subseteq \mathrm{Op}\cdot\mathcal{C}$.*

Observation 5.2.6 (Monotonicity). *Let $\mathcal{C}$ and $\mathcal{D}$ be two communication complexity classes such that $\mathcal{C} \subseteq \mathcal{D}$. Then for every operator $\mathrm{Op} \in \{\mathrm{co}, \forall, \exists, \mathrm{Mod}_k, \oplus, \mathrm{BP}\}$ we have $\mathrm{Op}\cdot\mathcal{C} \subseteq \mathrm{Op}\cdot\mathcal{D}$*

Observation 5.2.7 (Idempotency). *Let $\mathcal{C}$ be a communication complexity class that is closed under many-one reductions. Then for every operator $\mathrm{Op} \in \{\forall, \exists, \oplus\}$ we have $\mathrm{Op}\cdot\mathrm{Op}\cdot\mathcal{C} = \mathrm{Op}\cdot\mathcal{C}$*

The idempotency of the BP-operator follows from its probability amplification property (Observation 5.2.4).

Observation 5.2.8 (Idempotency of BP$\cdot$). *We have $\mathrm{BP}\cdot\mathrm{BP}\cdot\mathcal{C} = \mathrm{BP}\cdot\mathcal{C}$ for every communication complexity class $\mathcal{C}$ closed under majority reductions.*

Observation 5.2.9 (co$\cdot$ vs. $\cdots$). *Let $\mathcal{C}$ be a communication complexity class. We have $\mathrm{co}\cdot\exists\cdot\mathcal{C} = \forall\cdot\mathrm{co}\cdot\mathcal{C}$, $\mathrm{co}\cdot\forall\cdot\mathcal{C} = \exists\cdot\mathrm{co}\cdot\mathcal{C}$, and $\mathrm{co}\cdot\mathrm{BP}\cdot\mathcal{C} = \mathrm{BP}\cdot\mathrm{co}\cdot\mathcal{C}$.*

Definition 5.2.10 (Intersection & union). *Let $\mathcal{C}$ and $\mathcal{D}$ be communication complexity classes. The class $\mathcal{C}$ is* closed under $\mathcal{D}$-intersection *iff for all $A \in \mathcal{C}$ and $B \in \mathcal{D}$ we have $A \cap B \in \mathcal{C}$, and it is* closed under $\mathcal{D}$-union *iff for all $A \in \mathcal{C}$ and $B \in \mathcal{D}$ we have $A \cup B \in \mathcal{C}$.*

Definition 5.2.11 (Normal class). *We call a communication complexity class $\mathcal{C}$* normal *iff it is closed under $\mathbf{P}^{cc}$-intersection, $\mathbf{P}^{cc}$-union, and many-one reductions, and if it contains $\mathbf{P}^{cc}$.*

Observation 5.2.12 (co$\cdot$ vs. $\oplus\cdot$). *For a normal communication complexity class $\mathcal{C}$ we have $\mathrm{co}\cdot\oplus\cdot\mathcal{C} = \oplus\cdot\mathcal{C}$.*

Proof. Let $L \in \oplus\cdot\mathcal{C}$. There exist a bound $p \in \mathbf{poly}$ and a language $L_1 \in \mathcal{C}$ such that $L = \oplus^p(L_1)$. Define

$$L_2 := \{(\langle x, b_1 w_1\rangle, \langle y, b_2 w_2\rangle) \mid b_1, b_2 \in \mathbb{B}, (\langle x, w_1\rangle, \langle y, w_2\rangle) \in L_1\} \ ,$$
$$L_3 := \{(\langle x, 1 w_1\rangle, \langle y, 1 w_2\rangle) \mid |x| = |y| =: n, |w_1| = |w_2| = \lceil p(\log n)\rceil\} \ ,$$
$$L_4 := \{(\langle x, 0 w_1\rangle, \langle y, 0 w_2\rangle) \mid |x| = |y| =: n, w_1 = w_2 = 0^{\lceil p(\log n)\rceil}\} \ .$$

Then $L_2$ is in $\mathcal{C}$, because $\mathcal{C}$ is closed under many-one reductions, and $L_3, L_4 \in \mathbf{P}^{cc}$. The language $L_5 := (L_2 \cap L_3) \cup L_4$ is in $\mathcal{C}$, because $\mathcal{C}$, as a normal class, is closed under $\mathbf{P}^{cc}$-intersection and $\mathbf{P}^{cc}$-union. Define $L' := \oplus^{p+1}(L_5)$. Clearly, $\overline{L} = L' \in \oplus\cdot\mathcal{C}$. □

Observation 5.2.13. *If $\mathcal{C}$ is a communication complexity class closed under conjunctive reductions, then $\oplus \cdot \mathcal{C}$ is closed under conjunctive reductions.*

Using Observations 5.2.12 and 5.2.13 one can prove the result of Papadimitriou & Zachos (1983) in the setting of communication complexity as the one for the Turing machine model. by *e. g.* translating the proof in (Köbler *et al.* 1993, p. 125, Proposition 4.8).

Fact 5.2.14 (Papadimitriou & Zachos). *Let $\mathcal{C}$ be a normal communication complexity class closed under conjunctive reductions. Then $\oplus\mathbf{P}^{cc}(\oplus \cdot \mathcal{C}) = \oplus \cdot \mathcal{C}$.*

Swapping lemmata are well-known in the field of structural complexity theory. Below, we give a proof of a lemma of this type for the sake of completeness. The main ingredient is the probability amplification property of the BP-operator (Observation 5.2.4).

Lemma 5.2.15 (Swapping). *Let $\mathcal{C}$ be a communication complexity class closed under majority reductions. Then $\oplus \cdot \mathrm{BP} \cdot \mathcal{C} \subseteq \mathrm{BP} \cdot \oplus \cdot \mathcal{C}$.*

Proof. Let $L$ be a language in $\oplus \cdot \mathrm{BP} \cdot \mathcal{C}$. Then there exist a language $L'$ in $\mathrm{BP} \cdot \mathcal{C}$ and a bound $p' \in \mathbf{poly}$ such that $L = \oplus^{p'}(L')$. As $L' \in \mathrm{BP} \cdot \mathcal{C}$ and $\mathcal{C}$ is closed under majority reductions we use probability amplification to obtain a language $L''$ in $\mathcal{C}$ and a bound $p'' \in \mathbf{poly}$ such that

$$(\langle x,w\rangle,\langle y,w\rangle) \in L' \implies \Pr_r\left[(\langle\langle x,w\rangle,r\rangle,\langle\langle y,w\rangle,r\rangle) \in L''\right] \geq 1 - 2^{-l'_n-2} \text{ , and}$$
$$(\langle x,w\rangle,\langle y,w\rangle) \notin L' \implies \Pr_r\left[(\langle\langle x,w\rangle,r\rangle,\langle\langle y,w\rangle,r\rangle) \in L''\right] \leq 2^{-l'_n-2} \ .$$

for every $n$-bit input pair $(x,y)$ and witness $w$. Here, $l'_n := \lceil p'(\log n)\rceil$, and the random string $r$ is uniformly drawn from $\mathbb{B}^{l''_n}$, where $l''_n := \lceil p''(\log n)\rceil$. We define $W_{(x,y)} := \{w \in \mathbb{B}^{l'_n} \mid (\langle x,w\rangle,\langle y,w\rangle) \in L'\}$ and $\mathrm{Good}_n := \bigcap_{w\in\mathbb{B}^{l'_n}} \mathrm{Good}_{n,w}$, where $\mathrm{Good}_{n,w} := \{r \in \mathbb{B}^{l''_n} \mid \forall (x,y) \in (\mathbb{B}^n)^2 \colon (\langle\langle x,w\rangle,r\rangle,\langle\langle y,w\rangle,r\rangle) \in L'' \iff w \in W_{(x,y)}\}$. For a fixed $w_0$ we get

$$\Pr_r[r \notin \mathrm{Good}_n] \leq 2^{l'_n} \cdot \Pr_r[r \notin \mathrm{Good}_{n,w_0}] \leq 2^{l'_n} \cdot 2^{-l'_n-2} \leq 1/4 \ .$$

Thus, $\Pr_r[r \in \mathrm{Good}_n] \geq 3/4$. The language

$$L''' := \{(\langle\langle x,r\rangle,w\rangle,\langle\langle y,r'\rangle,w'\rangle) \mid (\langle\langle x,w\rangle,r\rangle,\langle\langle y,w'\rangle,r'\rangle) \in L''\}$$

is in $\mathcal{C}$ by closure under many-one reductions. In case $(x,y) \in L$ we have

$$\begin{aligned}
&\Pr_r\left[(\langle x,r\rangle,\langle y,r\rangle) \in \oplus^{p'}(L''')\right] \\
&= \Pr_r[|\{w \mid (\langle\langle x,w\rangle,r\rangle,\langle\langle y,w\rangle,r\rangle) \in L''\}| \text{ odd }] \\
&\geq \Pr_r\left[\forall w\colon w \in W_{(x,y)} \iff (\langle\langle x,w\rangle,r\rangle,\langle\langle y,w\rangle,r\rangle) \in L''\right] \qquad (5.2.16) \\
&\geq \Pr_r\left[\forall (x,y)\colon \forall w\colon w \in W_{(x,y)} \iff (\langle\langle x,w\rangle,r\rangle,\langle\langle y,w\rangle,r\rangle) \in L''\right] \\
&= \Pr_r[r \in \mathrm{Good}_n] \geq 3/4 \ ,
\end{aligned}$$

where (5.2.16) follows from $((x,y) \in L \iff |W_{(x,y)}|$ is odd). The case $(x,y) \notin L$ is treated similarly. We conclude $L \in \mathrm{BP} \cdot \oplus \cdot \mathcal{C}$. □

## 5.3 Valiant-Vazirani-Lemma

The Lemma of Valiant & Vazirani (1986) is a classical result in structural complexity theory. Valiant and Vazirani observed that if one randomly (using randomness ($R$)) adds certain clauses $\psi(R)$ to a satisfiable SAT-formula $\phi$, then with non-negligible probability $\phi \wedge \psi(R)$ has a unique satisfying assignment. Because "1" is an odd number and SAT is complete for the class **NP**, we can rephrase the statement in terms of complexity classes:

LEMMA 5.3.1 (Valiant & Vazirani). $\mathbf{NP} \subseteq \mathrm{RP} \cdot \oplus\mathbf{P}$.

Here, $\mathrm{RP} \cdot \mathcal{C}$ denotes the closure of $\mathcal{C}$ under randomized many-one reductions with one-sided error.

Is it possible to make an analogous statement in the setting of communication complexity? Of course, it is: the set intersection function, SI, and the inner product function mod 2, IP, correspond to SAT and $\oplus$SAT, respectively.

DEFINITION 5.3.2 (Set intersection). *The* set intersection function *is defined as* $\mathrm{SI} := (\mathrm{SI}_n)_{n\in\mathbb{N}}$, *where* $\mathrm{SI}_n(x,y) := [\,\exists i \in [n] \colon x_i = y_i = 1\,]$ *for* $x = x_1 \cdots x_n$ *and* $y = y_1 \cdots y_n$.

On inputs $x = x_1 \cdots x_n$ and $y = y_1 \cdots y_n$ Alice and Bob randomly reduce $\mathrm{SI}_n$ to $\mathrm{IP}_n$ as follows: first of all, they randomly choose a natural number $k$. The "right" $k$ would obey $2^{k-2} \leq |S| \leq 2^{k-1}$, where $S := \{i \in [n] \mid x_i = y_i = 1\}$. Then they randomly choose a pairwise independent hash function $h \colon [n] \to \{0,1\}^k$ that selects a subset $S_h := \{i \in [n] \mid h(i) = 0^k\}$ of the indices $[n]$. They call $\mathrm{IP}_n$ on $x' = x'_1 \cdots x'_n$ and $y' = y'_1 \cdots y'_n$, where $x'_i := x_i$ for $i \in S_h$, and 0 otherwise; analogously for $y'_i$. With non-negligible probability there is a unique index $i \in S_h$ satisfying $x'_i = y'_i = 1$. Thus, we have obtained

LEMMA 5.3.3 (Valiant & Vazirani). $\mathbf{NP}^{cc} \subseteq \mathrm{RP} \cdot \oplus\mathbf{P}^{cc}$.

As we have seen, it was no problem to prove a Valiant-Vazirani-Lemma in communication complexity. But what about the relativized version?

OPEN QUESTION 5.3.4. *Let $A$ be a language. Do we have*

$$\mathbf{NP}^{cc}(A) \subseteq \mathrm{RP} \cdot \oplus\mathbf{P}^{cc}(A) \ ?$$

Relativization seems to destroy the possibility to construct an efficient reduction. Let $\Pi^A := (\Pi^A_n)_{n\in\mathbb{N}}$ be an oracle protocol family for a language $L \in \mathbf{NP}^{cc}(A)$. Then $\Pi^A_n$ may have $2^{\mathrm{polylog}(n)}$ many oracle nodes. Thus, the different oracle answers might lead to $2^{2^{\mathrm{polylog}(n)}}$ many different partitions of the input space. A simple many-one reduction via characteristic vectors does not seem to work.

This problem can be circumvented by the use of complexity class operators. We will prove Toda's Theorem in the setting of communication complexity in Section 5.5 via the respective complexity class operators and the following *operator-theoretical version* of the Valiant-Vazirani-Lemma.

LEMMA 5.3.5 (Valiant & Vazirani). *Let $\mathcal{C}$ be a normal communication complexity class closed under conjunctive reductions. Then $\exists \cdot \mathcal{C} \subseteq \mathrm{BP} \cdot \oplus \cdot \mathcal{C}$.*

PROOF. The proof is an adaptation of an algebraic proof due to Fortnow in (Fortnow 1997, p. 88, Lemma 3.12): Let $L$ be a language in $\exists \cdot \mathcal{C}$. There exist a language $L' \in \mathcal{C}$ and a bound $p \in \mathbf{poly}$ such that $L = \exists^p(L')$. Define $l_n := \lceil p(\log n) \rceil$. We fix an input $(x,y) \in L$, $|x| = |y| = n$. Let $S := \{w \in \mathbb{B}^{l_n} \mid (\langle x, w\rangle, \langle y, w\rangle) \in L'\}$ be the set of witnesses of $(x,y)$ and $d := |S|$ its size. We pick a natural number $m$ such that

$\log(2l_nd) < m \leq \log(4l_nd)$ and encode the witnesses as polynomials over $F := \mathbb{F}_{2^m}$, the finite field with $2^m$ elements. We then consider pairs $(a, b) \in F^2$ and show that for a sizable fraction of them there will be exactly one polynomial $p$ representing a witness such that $p(a) = b$. The statement follows by choosing $m$, $a$ and $b$ at random. For a string $s = s_1 \cdots s_l$ we define the polynomial $p_s(X) := \sum_{i=1}^{l} s_i X^{i-1}$. We fix a witness $w$ in $S$. An element $a$ of $F$ is called $w$-good, if for all witnesses $w' \neq w$ in $S$ we have $p_w(a) \neq p_{w'}(a)$. Since $p_w$ and $p_{w'}$ can agree on at most $l_n$ elements, there are at least $|F| - l_nd$ many $w$-good elements in $F$. Consider the set $A_w$ containing all pairs $(a, p_w(a))$ for $w$-good elements $a$. The sets $A_w$ and $A_{w'}$ are disjoint for different strings $w$ and $w'$. Define $A := \bigcup_{w \in S} A_w$. Then $|A| \geq d(|F| - l_nd)$. We define the language $L''$ in $\mathcal{C}$ by

$$L'' := \{(\langle\langle x, r\rangle, w\rangle, \langle\langle y, r\rangle, w\rangle) \mid n := |x| = |y|, r = \langle m^*, a, b\rangle, m^* \in [2l_n], \\ a, b \in \mathbb{F}_{2^{m^*}}, |w| = l_n, p_w(a) = b, (\langle x, w\rangle, \langle y, w\rangle) \in L'\} \ ,$$

where $r = \langle m^*, a, b\rangle$ means that we use $r$ as an encoding of a natural number $m^*$ and field elements $a$ and $b$. Furthermore, define $L''' := \oplus^p(L'') \in \oplus \cdot \mathcal{C}$.

If $(x, y) \notin L$ then for all $w$ and $r$ the pair $(\langle\langle x, r\rangle, w\rangle, \langle\langle y, r\rangle, w\rangle)$ is not in $L''$, and thus $(x, y) \notin L'''$.

If $(x, y) \in L$ then with probability $1/2l_n$ we have $m = m^*$ as $m \leq \log 4l_nd \leq 2l_n$. In case $m = m^*$ the size of $A$ is at least $l_nd^2$, the size of $F^2$ is at most $16l_n^2d^2$. If we choose $(a, b)$ at random in $F^2$ we have a $1/16l_n$ chance of being in $A$. Thus, for fixed input $(x, y)$ the probability of choosing $r$ at random such that $m = m^*$ and $(a, b) \in A$ is at least $1/32l_n^2$. In this case there is exactly one witness $w$ for $(\langle x, r\rangle, \langle y, r\rangle)$ showing $(x, y) \in L'''$.

The class $\oplus \cdot \mathcal{C}$ is closed under majority reductions by Fact 5.2.14. Thus, probability amplification is possible, and we get $L \in \mathrm{BP} \cdot \oplus \cdot \mathcal{C}$. □

## 5.4 A protocol with few alternations for the inner product function mod 2

In this section we want to develop an alternating protocol with few alternations for the inner product function mod 2.

Definition 5.4.1 (Inner product function mod 2). *The* inner product function mod 2, $\mathrm{IP} := (\mathrm{IP}_n)_{n \in \mathbb{N}}$, *is defined as* $\mathrm{IP}_n(x, y) := \sum_{i \in [n]} x_i y_i \mod 2$, *where* $x = x_1 \cdots x_n$ *and* $y = y_1 \cdots y_n$.

For the moment, let $L_{\mathrm{IP}}$ denote the language corresponding to IP. It is complete for the class $\oplus\mathbf{P}^{cc}$ under many-one reductions. This is one of many reasons why the inner product function mod 2 has been studied extensively:

In (Kushilevitz & Nisan 1997, p. 12, Exercise 1.25) it was shown that $\mathrm{R}_0(\mathrm{IP}_n) \geq \mathrm{N}^0(\mathrm{IP}_n) \geq n - 1$ using the rectangle size method. This implies $L_{\mathrm{IP}} \notin \mathbf{coNP}^{cc}$. The lower bound $\mathrm{R}_0^{\mathrm{pub}}(\mathrm{IP}_n) \geq n - 1$ for the public coin model was shown in (Dietzfelbinger & Wunderlich 2007, p. 249, Example 3.7).

The distributional communication complexity of IP was studied in Chor & Goldreich (1988) improving on a result of Vazirani (1987). See also (Babai *et al.* 1986, p. 345, Lemma 9.3, Corollary 9.4). A proof similar to Chor & Goldreich (1988) was given in (Kushilevitz & Nisan 1997, p. 39, Example 3.29; p. 40, Exercise 3.30) that shows $\mathrm{R}_{\frac{1}{2}-\epsilon}(\mathrm{IP}_n) \geq n - \mathcal{O}(\log \frac{1}{\epsilon})$ using the discrepancy method. This implies $L_{\mathrm{IP}} \notin \mathbf{BPP}^{cc}$. Klauck (2003) showed a strong connection between majority covers and the discrepancy method. Thus, the result above actually gives $\mathrm{PP}(\mathrm{IP}_n) = \Theta(n)$. This implies $L_{\mathrm{IP}} \notin \mathbf{PP}^{cc}$. In the work of Forster (2002), a linear lower bound was established in the unbounded error communication complexity model, implying even $L_{\mathrm{IP}} \notin \mathbf{UPP}^{cc}$.

LEMMA 5.4.2. *Let $x = x_1 \cdots x_n$ and $y = y_1 \cdots y_n$ be inputs. Divide them into an odd number $k$ of blocks, i. e. $x = x^{(1)} \cdots x^{(k)}$ and $y = y^{(1)} \cdots y^{(k)}$. Then for $b \in \{0,1\}$ the following are equivalent:*

*(i)* $\mathrm{IP}_n(x,y) = b$.

*(ii) There exists an odd number $S \subseteq [k]$ of blocks such that $\mathrm{IP}(x^{(i)}, y^{(i)}) = b$ for $i \in S$ and $\mathrm{IP}(x^{(j)}, y^{(j)}) = 1 - b$ for $j \in \overline{S}$.*

PROOF. $(ii) \implies (i)$: The cardinality of $\overline{S}$ is even. Thus, we have

$$\begin{aligned}\mathrm{IP}_n(x,y) &= \sum_{i \in [k]} \mathrm{IP}(x^{(i)}, y^{(i)}) \mod 2 \\ &= \sum_{i \in S} \mathrm{IP}(x^{(i)}, y^{(i)}) + \sum_{j \in \overline{S}} \mathrm{IP}(x^{(j)}, y^{(j)}) \mod 2 \\ &= |S| \cdot b + |\overline{S}| \cdot (1-b) \mod 2 = b \ .\end{aligned}$$

$(i) \implies (ii)$: Define $S := \{i \in [k] \mid \mathrm{IP}(x^{(i)}, y^{(i)}) = b\}$. By the assumption, we have $b = \mathrm{IP}_n(x,y) = |S| \cdot b + |\overline{S}| \cdot (1-b) \mod 2$. If $b = 0$ then $|\overline{S}|$ is even, implying $|S|$ odd. If $b = 1$ then $|S| \mod 2 = 1$. □

The simple lemma above leads to a "divide and conquer"-strategy to compute the inner product function mod 2 with few alternations. This is implemented in Protocol $I_k(s,t,b)$ (Algorithm 1).

OBSERVATION 5.4.3. *On $n$-bit inputs $x = x_1 \cdots x_n$ and $y = y_1 \cdots y_n$ the protocol*

$$I_k(s,t,b) \text{ accepts} \iff \mathrm{IP}_{t-s+1}(x_s \cdots x_t, y_s \cdots y_t) = b \ .$$

*Thus, the protocol $I_k(1,n,1)$ computes $\mathrm{IP}_n(x,y)$.*

PROOF. The correctness of the protocol follows from Lemma 5.4.2 by induction on $t - s + 1$. □

There are two alternations in each round of the protocol, and the number of rounds is bounded by $t = \log n / \log k$. If we choose an odd natural number $k$ of size $(\log n)^{\mathcal{O}(1)}$, then the communication cost in each round is $\mathcal{O}(k)$ bits, and the number of alternations is $\mathcal{O}(\log n / \log\log n)$, substantially less than allowed. Recall that $\mathbf{PSPACE}^{cc}$ was defined as the class of languages which can be recognized with protocols using $(\log n)^{\mathcal{O}(1)}$ communication and $(\log n)^{\mathcal{O}(1)}$ many alternations. Especially, the number of alternations is allowed to be proportional to the communication cost.

We consider this as some evidence that the class $\oplus\mathbf{P}^{cc}$ is much "easier" than the class $\mathbf{PSPACE}^{cc}$, because the $\oplus\mathbf{P}^{cc}$-complete problem $L_{\mathrm{IP}}$ needs so few alternations. Finally, we conjecture that even the class $\mathrm{BP} \cdot \oplus\mathbf{P}^{cc}$ is much "easier" than $\mathbf{PSPACE}^{cc}$, because Schöning's generalization $\mathrm{BP} \cdot \mathcal{C} \subseteq \exists \cdot \forall \cdot \mathcal{C} \cap \forall \cdot \exists \cdot \mathcal{C}$ of the classical result of Lautemann, which is easily transferred into the communication complexity context, tells us that randomization with bounded error can be replaced with just two additional alternations.

## 5.5 Toda's Theorems

In this section we want to prove Toda's remarkable theorems (see Toda 1991) in the setting of communication complexity. This result was claimed by Lokam (2001) without proof. We give a definition of the polynomial hierarchy suitable for these purposes based on the complexity class operators defined in Section 5.2. Note that this definition of the polynomial hierarchy is equivalent to the one given in Babai *et al.* (1986) and the one given in Section 5.1.

**Algorithm 1**: Protocol $I_k(s,t,b)$

**Input**: Alice has $x = x_1 \cdots x_n$ and Bob has $y = y_1 \cdots y_n$
**Data**: Both know $s$, $t$, $b$ and the odd natural number $k$
**if** $(k \geq t - s + 1)$ **then**
  **begin**
    `/* Trivial protocol: Alice sends her input; both compute the value by themselves. */`
    Alice and Bob compute $b' := \mathrm{IP}_{t-s+1}(x_s \cdots x_t, y_s \cdots y_t)$ using the trivial protocol;
    `/* They return 1, if` $b$ `equals` $b'$`, and 0 otherwise: */`
    **return** $(b == b')$;
  **end**
**else**
  **begin**
    `/* Alice guesses the following strings and sends them to Bob: */`
    **Guess existentially** $S \subseteq [k], |S|$ *odd*;
    **Guess universally** $i \in S$;
    **Guess universally** $j \in \overline{S}$;
    **Guess universally** $h \in \{i, j\}$;
    `/* Both compute for themselves (no communication)`

$$d := t - s + 1 \ , \qquad s_1 := s + (h-1) \cdot B \ ,$$
$$B := \lceil d/k \rceil \ , \qquad t_1 := \min\{t, h \cdot B\} \ ,$$
$$b_1 := \begin{cases} b & , h = i \ , \\ 1 - b & , h = j \ . \end{cases}$$

    `*/`
    **return** $I_k(s_1, t_1, b_1)$;
  **end**
**end**

Definition 5.5.1 (Polynomial hierarchy). *The polynomial hierarchy* $\mathbf{PH}^{cc}$ *is defined as* $\mathbf{PH}^{cc} := \bigcup_{k\geq 0} \Sigma_k^{cc}$, *where each level is defined as*

$$\Sigma_0^{cc} := \mathbf{P}^{cc} \ ,$$
$$\Sigma_{k+1}^{cc} := \exists \cdot \mathrm{co} \cdot \Sigma_k^{cc} \ .$$

Toda's First Theorem 5.5.2. $\mathbf{PH}^{cc} \subseteq \mathrm{BP} \cdot \oplus \cdot \mathbf{P}^{cc}$.

Proof. The proof is analogous to the one in the Turing machine setting. We prove $\Sigma_k^{cc} \subseteq \mathrm{BP} \cdot \oplus \cdot \mathbf{P}^{cc}$ by induction on $k$:
*Case* $k = 0$: The class $\mathbf{P}^{cc}$ is closed under many-one reductions. The class $\oplus \cdot \mathbf{P}^{cc}$ is also closed under many-one reductions by Fact 5.2.14, because $\mathbf{P}^{cc}$ is closed under $\mathbf{P}^{cc}$-intersection, $\mathbf{P}^{cc}$-union, and conjunctive reductions. Thus, $\Sigma_0^{cc} = \mathbf{P}^{cc} \subseteq \oplus \cdot \mathbf{P}^{cc} \subseteq \mathrm{BP} \cdot \oplus \cdot \mathbf{P}^{cc}$ by the inclusion property of the $\oplus$- and BP-operator (Observation 5.2.5), respectively.
*Case* $k \to k+1$: We have

$$\Sigma_{k+1}^{cc} = \exists \cdot \mathrm{co} \cdot \Sigma_k^{cc} \tag{5.5.3}$$
$$\subseteq \exists \cdot \mathrm{co} \cdot \mathrm{BP} \cdot \oplus \cdot \mathbf{P}^{cc} \tag{5.5.4}$$
$$= \exists \cdot \mathrm{BP} \cdot \mathrm{co} \cdot \oplus \cdot \mathbf{P}^{cc} \tag{5.5.5}$$
$$= \exists \cdot \mathrm{BP} \cdot \oplus \cdot \mathbf{P}^{cc} \tag{5.5.6}$$
$$\subseteq \mathrm{BP} \cdot \oplus \cdot \mathrm{BP} \cdot \oplus \cdot \mathbf{P}^{cc} \tag{5.5.7}$$
$$\subseteq \mathrm{BP} \cdot \mathrm{BP} \cdot \oplus \cdot \oplus \cdot \mathbf{P}^{cc} \tag{5.5.8}$$
$$= \mathrm{BP} \cdot \mathrm{BP} \cdot \oplus \cdot \mathbf{P}^{cc} \tag{5.5.9}$$
$$= \mathrm{BP} \cdot \oplus \cdot \mathbf{P}^{cc} \ . \tag{5.5.10}$$

(5.5.3) By Definition 5.5.1.

(5.5.4) By the induction hypothesis for $\Sigma_k^{cc}$ and monotonicity (Observation 5.2.6) of the operators co· and $\exists$·.

(5.5.5) By Observation 5.2.9.

(5.5.6) By closure under complement of $\oplus \cdot \mathbf{P}^{cc}$ (Observation 5.2.12).

(5.5.7) By the Valiant-Vazirani-Lemma (Lemma 5.3.5). Its application is possible, because $\mathrm{BP} \cdot \oplus \cdot \mathbf{P}^{cc}$ is normal and closed under conjunctive reductions.

(5.5.8) By the Swapping-Lemma (Lemma 5.2.15) and monotonicity of the BP-operator (Observation 5.2.6). The Swapping-Lemma can be applied, because $\oplus \cdot \mathbf{P}^{cc}$ is closed under majority reductions.

(5.5.9) By idempotency of the $\oplus$-operator (Observation 5.2.7).

(5.5.10) By idempotency of the BP-operator (Observation 5.2.8). This holds because $\oplus \cdot \mathbf{P}^{cc}$ is closed under majority reductions.

□

For the Turing machine model the fact below was established in Angluin (1980).

Fact 5.5.11 (Angluin). $\mathbf{P}^{cc}(\mathbf{PP}^{cc}) = \mathbf{P}^{cc}(\#\mathbf{P}^{cc})$.

Proof. The proof is analogous to the one in the Turing machine setting. Alice and Bob can compute every $\#\mathbf{P}^{cc}$-function $f$ by binary search with polylog communication asking oracle queries to $\mathrm{Graph}_{\leq}(f) \in \mathbf{PP}^{cc}$, where

$$\mathrm{Graph}_{\leq}(f) := \{(\langle x, v\rangle, \langle y, v\rangle) \mid (v)_2 \leq f(x, y)\} \ ,$$

and $(v)_2$ is the binary value of the string $v$. □

TODA'S SECOND THEOREM 5.5.12. $\mathrm{BP} \cdot \oplus\mathbf{P}^{cc} \subseteq \mathbf{P}^{cc}(\#\mathbf{P}^{cc})$.

PROOF. The proof is analogous to the one in the Turing machine setting. If $\Pi := (\Pi_n)_{n\in\mathbb{N}}$ is an efficient family of counting protocols with $\mathrm{acc}_\Pi := (\mathrm{acc}_{\Pi_n})_{n\in\mathbb{N}}$ in $\#\mathbf{P}^{cc}$, and if we choose $p \in \mathbf{poly}$, then there exists an efficient family of counting protocols $\Pi' := (\Pi'_n)_{n\in\mathbb{N}}$ such that $\mathrm{acc}_{\Pi'_n}(x,y) = (1 + \mathrm{acc}_{\Pi_n}(x,y)^{\lceil p(\log n)\rceil})^{\lceil p(\log n)\rceil}$, and $\mathrm{acc}_{\Pi'}$ is in $\#\mathbf{P}^{cc}$, because the class $\#\mathbf{P}^{cc}$ contains all constant functions and is closed under addition and multiplication. □

We close this section with a corollary summing up the previous results.

COROLLARY 5.5.13.

$$\mathbf{PH}^{cc} \subseteq \mathrm{BP} \cdot \oplus\mathbf{P}^{cc} \subseteq \mathbf{P}^{cc}(\#\mathbf{P}^{cc}) = \mathbf{P}^{cc}(\mathbf{PP}^{cc}) \subseteq \mathbf{PSPACE}^{cc} \ . \tag{5.5.14}$$

## 5.6 Approximate rank

In this section we define notions of *approximate* $\mathbb{F}$*-rank* for fields $\mathbb{F}$, which to the author's knowledge have not been defined before. Especially, we are interested in approximate $\mathbb{F}_2$-rank and approximate $\mathbb{R}$-rank, because the logarithm of the first one characterizes the BP-Parity-P complexity and the logarithm of the second one is a lower bound method for bounded error randomized communication complexity.

DEFINITION 5.6.1 (Approximate $\mathbb{F}$-rank). *Let $\mathbb{F}$ be a field, let $M$ be a Boolean matrix with row set $\mathcal{X}$ and column set $\mathcal{Y}$, let $\mu$ be a probability distribution on $\mathcal{X}\times\mathcal{Y}$, and let $\epsilon \geq 0$ be a real number. The* $(\mu,\epsilon)$-approximate $\mathbb{F}$-rank of $M$ *is defined as*

$$\mathbb{F}\text{-rank}^{\mu}_{\epsilon}(M) := \min\{\mathbb{F}\text{-rank}(\tilde{M}) \mid \mu(\tilde{M} \neq M) \leq \epsilon, \tilde{M} \text{ a Boolean matrix}\} \ .$$

*Here,* $\mu(\tilde{M} \neq M) := \mu\{(x,y) \mid \tilde{M}_{x,y} \neq M_{x,y}\}$.
*The* $\epsilon$-approximate $\mathbb{F}$-rank of $M$ *is defined as*

$$\mathbb{F}\text{-rank}^{*}_{\epsilon}(M) := \max_{\mu} \mathbb{F}\text{-rank}^{\mu}_{\epsilon}(M) \ .$$

EXAMPLE 5.6.2. Let $U$ denote a uniform distribution. We want to give two small examples.

(i) Consider the following matrix:

$$M := \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \ .$$

We can prove that

$$\mathbb{R}\text{-rank}^{U}_{1/9}(M) = 3 \ ,$$

because for every $x \in \{0,1\}$ we have

$$\begin{vmatrix} x & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 1 & 1 \\ 1 & x & 1 \\ 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & x \end{vmatrix} = 2 - x$$

and

$$\begin{vmatrix} 0 & x & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 1 & x \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 1 & 1 \\ x & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 1 & 1 \\ 1 & 0 & x \\ 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & x & 0 \end{vmatrix} = \begin{vmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & x \end{vmatrix} = 1 + x \ .$$

(ii) We do not have such a stable behavior for matrices over the field $\mathbb{F}_2$: For every $n \times n$-matrix $N$ over $\mathbb{F}_2$ with full rank $n$ we have

$$\mathbb{F}_2\text{-rank}^U_{1/n^2}(N) = n - 1 \ .$$

This can be seen by looking at the determinant of $N$. First-row-expansion gives

$$1 = |N| = \bigoplus_{i=1}^{n} N_{1,i} \cdot \left| N^{(1,i)} \right|$$

Of course, $N^{(1,i)}$ denotes the matrix $N$ with row 1 and column $i$ deleted.

Thus, there exists an odd set $I$ of indices such that

$$N_{1,i} \cdot \left| N^{(1,i)} \right| = 1$$

for all $i \in I$. Complementing a single $(1, i_0)$-entry of $N$ for an $i_0 \in I$ reduces the rank by one. ◊

The next theorem shows that the logarithm of approximate $\mathbb{F}_2$-rank is a measure characterizing the BP-Parity-P complexity of a function.

THEOREM 5.6.3 (Characterization). *Let $f$ be a Boolean function, and let $\epsilon \geq 0$ be a real number. Then we have*

$$\log \mathbb{F}_2\text{-rank}^*_\epsilon(M^f) \leq \mathrm{BP}{\oplus}\mathrm{P}^{\mathrm{pub}}_\epsilon(f) \leq \log \mathbb{F}_2\text{-rank}^*_\epsilon(M^f) + \mathcal{O}(1) \ .$$

PROOF. For the lower bound,

$$\begin{aligned}
\mathrm{BP}{\oplus}\mathrm{P}^{\mathrm{pub}}_\epsilon(f) &= \max_\mu \oplus\mathrm{P}^\mu_\epsilon(f) && (5.6.4)\\
&= \max_\mu \min_{\tilde{f}\colon \mu(\tilde{f} \neq f) \leq \epsilon} \oplus\mathrm{P}(\tilde{f}) && (5.6.5)\\
&\geq \max_\mu \min_{\tilde{f}\colon \mu(\tilde{f} \neq f) \leq \epsilon} \log \mathbb{F}_2\text{-rank}(M^{\tilde{f}}) && (5.6.6)\\
&= \log \max_\mu \min_{\tilde{f}\colon \mu(\tilde{f} \neq f) \leq \epsilon} \mathbb{F}_2\text{-rank}(M^{\tilde{f}})\\
&= \log \max_\mu \min_{\tilde{f}\colon \mu(\tilde{M} \neq M^f) \leq \epsilon} \mathbb{F}_2\text{-rank}(\tilde{M})\\
&= \log \max_\mu \mathbb{F}_2\text{-rank}^\mu_\epsilon(M^f)\\
&= \log \mathbb{F}_2\text{-rank}^*_\epsilon(M^f) \ ,
\end{aligned}$$

where (5.6.4) holds by Observation 3.1.40, (5.6.5) by Observation 3.1.34, and (5.6.6) by Fact 3.3.7, respectively. The upper bound can be derived similarly using

$$\oplus\mathrm{P}(\tilde{f}) \leq \log \mathbb{F}_2\text{-rank}(M^{\tilde{f}}) + \mathcal{O}(1)$$

of Fact 3.3.7. □

The same argument with $\mathbb{R}$-rank shows

THEOREM 5.6.7. *Let $f$ be a Boolean function, and let $\epsilon \geq 0$ be a real number. Then we have*

$$\mathrm{R}^{\mathrm{pub}}_\epsilon(f) \geq \log \mathbb{R}\text{-rank}^*_\epsilon(M^f) \ .$$

Thus, we have obtained that the logarithm of the approximate $\mathbb{R}$-rank is a new lower bound method for bounded error randomized communication complexity. Comparisons between the logarithm of approximate $\mathbb{R}$-rank and other methods have not been made yet.

Note that if the logarithmic rank conjecture holds, then bounded error randomized communication complexity and the logarithm of approximate $\mathbb{R}$-rank are polynomially tight. So, we ask:

Open Question 5.6.8 (Optimality). *Is it true that none of the known lower bound methods developed for bounded error randomized communication complexity are better than the logarithm of approximate $\mathbb{R}$-rank up to a polynomial gap?*

## 5.7 Matrix rigidity

The concept of *(matrix) rigidity* was introduced by Valiant (1977) as a tool to derive lower bounds in circuit complexity. A matrix has high rigidity, if small perturbations, i. e. changes of a small number of entries in the matrix, do not lower the rank much. Proving a strong enough lower bound on the rigidity of a matrix implies a non-trivial lower bound, i. e. a superlinear size or a superlogarithmic depth, on the complexity of any linear circuit computing the set of linear forms associated with it. Although it has been shown that most matrices have high rigidity, despite considerable efforts by many researchers (see e. g. Cheraghchi (2005); Codenotti (2000); Codenotti *et al.* (2000); Friedman (1993); Lokam (2000, 2001); Midrijanis (2005); Pudlák (1994); Pudlák & Rödl (1994); Shokrollahi *et al.* (1997); de Wolf (2006)) no explicit construction of a rigid family of matrices over finite fields is known. For infinite fields Lokam (2006) was able to derive quadratic lower bounds for the rigidity of explicit matrix families using the concept of (generalized) Smolensky-Shoup-dimension.

In this section we establish an explicit connection between measures of communication complexity and matrix rigidity. Especially, quadratic lower bounds for rigidity translate to linear lower bounds in communication complexity. Prior to this work it was only known that high rigidity of the communication matrices of a concrete function family implies high $\mathbf{AC}^0$-dimension, and thus yields a separation of the communication complexity classes $\mathbf{PH}^{cc}$ and $\mathbf{PSPACE}^{cc}$, see Lokam (2001) for details. To the best of the author's knowledge, no formula that yields a tight relationship between rigidity and a communication complexity measure was known before. We establish such a relationship here. Using the result of Valiant (1977) that most Boolean matrices over a finite field have high rigidity, together with the correspondence between the BP-Parity-P complexity and approximate $\mathbb{F}_2$-rank, we prove that most Boolean functions have high, i. e. $\Omega(n/\log n)$, BP-Parity-P complexity.

The formal definition of matrix rigidity is given below for the sake of completeness. We also introduce a variant defined for Boolean matrices we call *Boolean rigidity*, where the variation is only over Boolean matrices. In other words, only toggling values between 0 and 1 is allowed for rank reduction. In contrast, in the original definition of matrix rigidity, one can reduce the rank of a Boolean matrix by replacing zeros and ones with arbitrary real numbers.

Definition 5.7.1 (Rigidity). *Let $M$ be a matrix over a field $\mathbb{F}$. The* (matrix) rigidity $\mathrm{R}^{\mathbb{F}}_M$ of $M$ *is defined as*

$$\mathrm{R}^{\mathbb{F}}_M(r) := \min\{\mathrm{wt}(\tilde{M} - M) \mid \mathbb{F}\text{-rank}(\tilde{M}) \leq r, \tilde{M} \text{ a matrix over } \mathbb{F}\} \ ,$$

*i. e. the minimum number of entries in $M$ that must be changed in order to reduce the rank to $r$.*

Definition 5.7.2 (Boolean rigidity). *Let $M$ be a Boolean matrix over a field $\mathbb{F}$. The* Boolean rigidity $\mathrm{B}^{\mathbb{F}}_M$ of $M$ *is defined as*

$$\mathrm{B}^{\mathbb{F}}_M(r) := \min\{\mathrm{wt}(\tilde{M} - M) \mid \mathbb{F}\text{-rank}(\tilde{M}) \leq r, \tilde{M} \text{ a Boolean matrix } \} \ .$$

We observe that both rigidity and Boolean rigidity are monotonically decreasing functions, i. e.

$$r_1 \leq r_2 \implies \left(\mathrm{R}^{\mathbb{F}}_M(r_1) \geq \mathrm{R}^{\mathbb{F}}_M(r_2) \text{ and } \mathrm{B}^{\mathbb{F}}_{M'}(r_1) \geq \mathrm{B}^{\mathbb{F}}_{M'}(r_2)\right) \ , \tag{5.7.3}$$

where $M$ is a matrix over $\mathbb{F}$, and $M'$ is a Boolean matrix, respectively. In addition, we have $\mathrm{R}^{\mathbb{F}}_{M'}(r) \leq \mathrm{B}^{\mathbb{F}}_{M'}(r)$.

Let $M$ be a Boolean $n \times n$-matrix. Obviously, in case $\mu$ is a uniform distribution $U$, the close connection between approximate $\mathbb{F}_2$-rank and Boolean rigidity is as follows:

$$\mathbb{F}\text{-rank}^U_\epsilon(M) \leq r \iff \mathrm{B}^{\mathbb{F}}_M(r) \leq \epsilon \cdot n^2 \ . \tag{5.7.4}$$

In case $\mathbb{F} = \mathbb{F}_2$ rigidity and Boolean rigidity coincide, and we obtain

$$\mathbb{F}_2\text{-rank}^U_\epsilon(M) \leq r \iff \mathrm{R}^{\mathbb{F}_2}_M(r) \leq \epsilon \cdot n^2 \ . \tag{5.7.5}$$

In case $\mathbb{F} = \mathbb{R}$ we only obtain one direction

$$\mathbb{R}\text{-rank}^U_\epsilon(M) \leq r \implies \mathrm{R}^{\mathbb{R}}_M(r) \leq \epsilon \cdot n^2 \ . \tag{5.7.6}$$

The following corollary relates BP-Parity-P complexity with matrix rigidity, and comes in handy in the proof of Theorem 5.7.10 below.

Corollary 5.7.7. *Let $f\colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ be a Boolean function. Then for every $\epsilon > 0$ we have*

$$\epsilon \geq \frac{\mathrm{R}^{\mathbb{F}_2}_{M^f}(2^{\mathrm{BP}\oplus\mathrm{P}^{\mathrm{pub}}_\epsilon(f)})}{2^{2n}} \ .$$

Proof. Let $U$ be the uniform distribution on $\mathbb{B}^n \times \mathbb{B}^n$. Define

$$r_U := \mathbb{F}_2\text{-rank}^U_\epsilon(M^f) \leq 2^{\mathrm{BP}\oplus\mathrm{P}^{\mathrm{pub}}_\epsilon(f)} \ .$$

Then $\mathrm{R}^{\mathbb{F}_2}_{M^f}(r_U) \leq \epsilon \cdot 2^{2n}$ by (5.7.5). The result follows from (5.7.3). □

Again, the same argument with $\mathbb{R}$-rank shows

Corollary 5.7.8. *Let $f\colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ be a Boolean function. Then for every $\epsilon > 0$ we have*

$$\epsilon \geq \frac{\mathrm{R}^{\mathbb{R}}_{M^f}(2^{\mathrm{R}^{\mathrm{pub}}_\epsilon(f)})}{2^{2n}} \ .$$

In (Valiant 1977, p. 172–173, Theorem 6.4(ii)) showed that over a finite field most Boolean matrices have high rigidity:

Fact 5.7.9 (Valiant). *For all natural numbers $n$ and $r$ with $r < n - \sqrt{2n + \log n}$ a $(1 - 1/n)$-fraction of all Boolean $n \times n$-matrices $M$ has rigidity*

$$\mathrm{R}^{\mathbb{F}_2}_M(r) \geq \frac{(n-r)^2 - 2n - \log n}{2\log n + 1} \ .$$

Theorem 5.7.10. *For $n$ sufficiently large, a $(1-1/2^n)$-fraction of all Boolean functions $f\colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ has BP-Parity-P complexity*

$$\mathrm{BP}\oplus\mathrm{P}^{\mathrm{pub}}_{1/4}(f) \geq \Omega\left(\frac{n}{\log n}\right) \ .$$

Proof. There exists a constant $c$ such that for $n$ sufficiently large a $(1 - 1/2^n)$-fraction of all Boolean functions $f\colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ has rigidity

$$\mathrm{R}^{\mathbb{F}_2}_{M^f}(r) \geq c \cdot \frac{(2^n - r)^2}{n} \ ,$$

if $r \leq 2^{n-1}$. Fix such a function $f$. Define $b(n) := \mathrm{BP}{\oplus}\mathrm{P}^{\mathrm{pub}}_{1/4}(f)$, $t(n) := 6 \cdot \log(2n/c)$, and $\epsilon(n) := \frac{1}{2} \cdot (\frac{3}{4})^{t(n)/2}$. By probability amplification (Fact 3.1.23), we have

$$\mathrm{BP}{\oplus}\mathrm{P}^{\mathrm{pub}}_{\epsilon(n)}(f) \leq t(n)b(n) \ .$$

Assume for a contradiction that $b(n) < \frac{n-1}{t(n)}$. Define $r(n) := 2^{\mathrm{BP}\oplus\mathrm{P}^{\mathrm{pub}}_{\epsilon(n)}(f)}$. Then by Corollary 5.7.7,

$$\frac{1}{2} \cdot (\frac{3}{4})^{t(n)/2} \geq \frac{\mathrm{R}^{\mathbb{F}_2}_{M^f}(r(n))}{2^{2n}} \geq \frac{c}{n} \cdot (1 - 2^{b(n)t(n)-n})^2 \geq \frac{c}{4n} \ ,$$

contradicting $(\frac{3}{4})^{t(n)/2} = (\frac{27}{64})^{\log(2n/c)} < (\frac{1}{2})^{\log(2n/c)} = \frac{c}{2n}$. We conclude

$$\mathrm{BP}{\oplus}\mathrm{P}^{\mathrm{pub}}_{1/4}(f) = b(n) \geq \frac{n-1}{t(n)} \ .$$

□

As mentioned at the beginning of this section, many researchers tried to prove high rigidity for explicit matrix families without success. Especially, they looked at Hadamard matrices. A Boolean Hadamard matrix is just the communication matrix of IP, the inner product function mod 2. (See Definition 5.4.1) The BP-Parity-P complexity of IP is low ($\oplus\mathrm{P}(\mathrm{IP}_n) \leq \log n + 2$). Thus, one cannot prove high rigidity for $M^{\mathrm{IP}_n}$ with techniques that can be applied to $\mathrm{R}^{\mathbb{F}_2}$.

## 5.8 Quasi-random graphs

We investigate a new connection between communication complexity and the fascinating field of *quasi-random graphs* (see e. g. Chung *et al.* 1989). We think that problems based on adjacency questions about quasi-random graph families have high BP-Parity-P complexity, and thus are good candidates for separating the polynomial hierarchy from polynomial space.

While Chung & Tetali (1993) have shown that high communication complexity leads to quasi-randomness, we prove that under certain conditions the opposite direction holds, too! Unfortunately, we cannot prove lower bounds for the BP-Parity-P complexity, but we are able to show that the Parity-P complexity of such problems is lower bounded by $\log(1/P(n)) - \mathcal{O}(1)$, where $P(n)$ denotes the edge density of the graph family. Thus, known constructions of sparse quasi-random graph families like Erdös-Renyi graphs (defined below) yield many explicit problems provably outside of the class Parity-P.

### 5.8.1 Basic definitions

Let $\mathcal{G} := (G_n)_{n \geq 1}$ be a $D$-regular family of graphs such that $G_n$ is a $D(n)$-regular graph on $N(n) = 2^n$ nodes. We define the *edge density of* $\mathcal{G}$ by

$$P(n) := \frac{|E(G_n)|}{\left|\binom{V(G_n)}{2}\right|} = 2 \cdot \frac{D(n)}{N(n) - 1} \ .$$

We only consider graph families with $P(n)N(n) \to \infty$ for $n \to \infty$.

Graph families define problems in communication complexity. For a graph $G$ let $\mathrm{EDGE}_G$ denote the Boolean function such that the communication matrix of $\mathrm{EDGE}_G$ equals the adjacency matrix of $G$. (In other words, Alice has $x \in V(G)$, Bob has $y \in V(G)$, and they want to know if $\{x, y\} \in E(G)$.) Then, a graph family $\mathcal{G} := (G_n)_{n \in \mathbb{N}}$ defines a family $\mathrm{EDGE}_{\mathcal{G}} := (\mathrm{EDGE}_{G_n})_{n \in \mathbb{N}}$ of Boolean functions.

Definition 5.8.1 (Discrepancy). *A graph family $\mathcal{G}$ has the* discrepancy property **DISC(1)**, *if for all subsets $X, Y \subseteq V(G_n)$ we have*

$$|e_{G_n}(X,Y) - P(n)|X||Y|| = o\left(P(n)(N(n))^2\right) \ .$$

Graph families with the discrepancy property have been thoroughly studied in the theory of *quasi-random graphs*. For space reasons, we do not make any attempt to give an introduction into this fascinating field but we refer the reader to e. g. Chung & Graham (2002) and Krivelevich & Sudakov (2006) as possible starting points.

From here on, we call a graph family *quasi-random*, if it has the discrepancy property.

A quasi-random graph family is *dense*, if $D(n) = \Theta(N(n))$, and *sparse*, if $D(n) = o(N(n))$.

## 5.8.2 Almost superregular problems

In our opinion, what makes sparse quasi-random graph families amenable to high lower bounds in communication complexity are their *superregularity properties*.

Definition 5.8.2 (Almost superregular). *Let $A, B\colon \mathbb{N} \to \mathbb{N}$ be functions, and let $M := (M_n)_{n\in\mathbb{N}}$ be a family of matrices $M_n\colon \mathcal{X}_n \times \mathcal{Y}_n \to \mathbb{F}$ over a field $\mathbb{F}$ such that $|\mathcal{X}_n| = |\mathcal{Y}_n| =: N(n)$. We call the family $M$* almost $(A, B)$-superregular over $\mathbb{F}$, *if for every $A(n) \times A(n)$-submatrix $K$ of $M_n$ we have $\mathbb{F}$-rank$(K) \geq B(n)$.*

*We call a function family $(f_n)_{n\in\mathbb{N}}$* almost $(A, B)$-superregular over $\mathbb{F}$ *if the corresponding family $(M^{f_n})_{n\in\mathbb{N}}$ of communication matrices is almost $(A, B)$-superregular over $\mathbb{F}$.*

Of course, the definition above only makes sense for $B \leq A$. *Superregular matrices (over $\mathbb{F}$)* were defined in Valiant (1977) as matrices such that every quadratic submatrix has full rank (over $\mathbb{F}$). Thus, for $N(n) := 2^n$, a family of superregular $N(n) \times N(n)$-matrices over $\mathbb{F}$ is almost $(A, A)$-superregular over $\mathbb{F}$ for every function $A$.

Unfortunately, in contrast to what was suggested (without proof) in (Hoory *et al.* 2006, p. 11–12), families of Boolean superregular matrices over a fixed finite field do not exist.

Theorem 5.8.3. *For every fixed finite field $\mathbb{F}$ there do not exist families of superregular matrices.*

Proof. First of all, we consider Boolean families over the field $\mathbb{F} = \mathbb{F}_2$. Assume that there exists a superregular family $(M_n)_{n\in\mathbb{N}}$ of Boolean $n \times n$-matrices. We define

$$\begin{aligned} K(n) &:= n \ , \\ N(n) &:= 2^{K(n)+2} \log K(n) \ . \end{aligned}$$

Choose an arbitrary natural number $n_0 \geq 21$. The matrix $W := M_{N(n_0)}$ is an edge coloring of the biclique $K_{N(n_0),N(n_0)}$. Bipartite Ramsey theory (see e. g. the result of Conlon (2008) or prior work) tells us that in this big biclique there exists a small biclique $K_{K(n_0),K(n_0)}$ that is monochromatic under $W$. This means that $W$ contains an $n_0 \times n_0$-submatrix $T$ consisting of zeros only, or ones only. Thus, $\mathbb{F}$-rank$(T) \in \{0, 1\}$ in contradiction to the superregularity property implying $\mathbb{F}$-rank$(T) = n_0 \geq 21$.

The same argument can be applied to matrices defined over an arbitrary fixed finite field $\mathbb{F}$. Then, one has to use $|\mathbb{F}|$-colorings instead of 2-colorings and the bound $N$ is higher. □

But almost $(A, B)$-superregular matrix families exist for certain functions $A$ and $B$ over every finite field. As the reader might have already guessed, such families are given by the adjacency matrices of sparse quasi-random graph families.

Almost superregularity over the field of real numbers can be elegantly proven via spectral techniques. For this, we define

Definition 5.8.4 (Matrix norms). *Let $A$ be a complex $n \times n$-matrix.*

*(i) The* spectral norm of $A$ *is defined as* $||A|| := \max_{x \neq 0} ||Ax||/||x||$.

*(ii) The* Frobenius norm of $A$ *is defined as* $||A||_F := \sqrt{\sum_{i,j} |A_{i,j}|^2}$.

Definition 5.8.5 (Hamming-weight). *For a matrix $M$ over a field $\mathbb{F}$ we define the* Hamming-weight of $M$, $\mathrm{wt}(M)$, *as the number of nonzero entries in $M$.*

Definition 5.8.6 (Approximate Hamming-weight). *Let $\theta > 0$ be a real number. For a Boolean $n \times n$-matrix $A$ we define the* $\theta$-approximate Hamming-weight of $A$, $\widetilde{\mathrm{wt}}_\theta(A)$, *as the minimum Hamming-weight of a $(\theta n) \times (\theta n)$-submatrix of $A$.*

In other words, we consider all $(\theta n) \times (\theta n)$-submatrices, count the number of ones in them, and take the minimum.

Lemma 5.8.7. *Let $f := (f_n)_{n \in \mathbb{N}}$, $f_n \colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$, be a family of Boolean functions. Then for every constant real number $\theta > 0$ the family $f$ is almost $\left(\theta N, \frac{\widetilde{\mathrm{wt}}_\theta(M^{f_n})}{||M^{f_n}||^2}\right)$-superregular over $\mathbb{R}$.*

Proof. A basic fact from linear algebra (see e. g. Lokam 2001) is that for every submatrix $B$ of a matrix $A$ we have $\mathbb{R}\text{-rank}(B) \geq ||B||_F^2/||A||^2$. Note that for a Boolean matrix $B$ we have $||B||_F^2 = \mathrm{wt}(B)$. □

Theorem 5.8.8. *Let $\mathcal{G} := (G_n)_{n \in \mathbb{N}}$ be a $D$-regular quasi-random graph family with edge density $P$. For every constant real number $\theta > 0$ the family* $\mathrm{EDGE}_{\mathcal{G}}$ *is almost $\left(\theta N, \Omega\left(\theta^2/P\right)\right)$-superregular over $\mathbb{R}$.*

Proof. Let $M_n := A^{G_n}$. First of all, $||M_n|| = D(n)$, because $G_n$ is $D(n)$-regular. Let $B$ be a $(\theta N(n)) \times (\theta N(n))$-submatrix of $M_n$ that realizes $\widetilde{\mathrm{wt}}_\theta(M_n)$. By the discrepancy property, we have

$$\begin{aligned} \mathrm{wt}(B) &= (1 + o(1)) \cdot P(n) \cdot (\theta N(n))^2 \\ &\approx 2\theta^2 D(n) N(n) \text{ , for } n \text{ large.} \end{aligned}$$

Applying Lemma 5.8.7 yields the lower bound. □

The next theorem shows that the result above also holds for the field $\mathbb{F}_2$. Of course, it is proved in a different way, because we do not have spectral techniques over $\mathbb{F}_2$.

Theorem 5.8.9. *Let $\mathcal{G} := (G_n)_{n \in \mathbb{N}}$ be a $D$-regular quasi-random graph family with edge density $P$. For every constant real number $\theta > 0$ the family* $\mathrm{EDGE}_{\mathcal{G}}$ *is almost $\left(\theta N, \Omega\left(\theta^2/P\right)\right)$-superregular over $\mathbb{F}_2$.*

Proof. Let $n$ be sufficiently large. Let $M_n := A^{G_n}$, and define $A := \theta N$. Consider an arbitrary $A(n) \times A(n)$-submatrix $T$ of $M_n$. We want to show that $T$ has a high $\mathbb{F}_2$-rank. Let $U$ and $V$ be the subsets of $V(G_n)$ of size $A(n)$ that correspond to the rows and columns of $T$, respectively. By the discrepancy property of $\mathcal{G}$ we have

$$e(U, V) \approx P(n) \cdot (A(n))^2 \approx 2\theta^2 D(n) N(n) \text{ .}$$

There exists a subset $V' \subseteq V$ of size $\approx 2\theta^2 N(n)$ such that $e(U, v) \geq 1$ for every $v \in V'$, because of

$$|V'| D(n) \geq e(U, V') = e(U, V) \approx 2\theta^2 D(n) N(n) \text{ .}$$

Let $T'$ be the submatrix of $T$, where the columns are restricted to $V'$. We successively permute rows and columns of $T'$ in order to obtain a "stair" of ones starting with the stairhead in the upper left corner and going down, where each stair has length $\leq D(n)$. Thus, the number of stairs is at least $\approx (2\theta^2 N(n))/D(n)$ implying that $B(n) := \mathbb{F}_2\text{-rank}(T) \geq \mathbb{F}_2\text{-rank}(T') \approx 2\theta^2 \frac{D(n)}{N(n)}$, for $n$ sufficiently large. We conclude that the family $\text{EDGE}_{\mathcal{G}}$ is almost $(A, B)$-superregular over $\mathbb{F}_2$.

Now, we permute $T'$: Take the first column $v_1 \in V'$. By definition of $V'$ there exists a row $u_1 \in U$ that is a neighbor of $v_1$. Take $u_1$ as the new first row. It has $t_1 \leq D(n)$ neighbors $v_1, \ldots, v_{t_1}$ in $V'$. Permute the columns such that these neighbors form the first $t_1$ columns of $T'$. We created the first stair. Now, take a column $v_{t_1+1} \in V' - \{v_1, \ldots, v_{t_1}\}$ and continue this process to create the next stairs. This can be done at least $|V'|/D(n)$ many times. □

### 5.8.3 Lower bounds

The results obtained in the last subsection yield strong lower bounds for worst case deterministic and parity communication complexity.

Given two function families $f := (f_n)_{n\in\mathbb{N}}$ and $g := (g_n)_{n\in\mathbb{N}}$, we call $f_n \colon \mathcal{X}'_n \times \mathcal{Y}'_n \to \mathcal{Z}_n$ a *large subfunction* of $g_n \colon \mathcal{X}_n \times \mathcal{Y}_n \to \mathcal{Z}_n$, if there exists a constant real number $\theta > 0$ such that $f_n$ is the restriction of $g_n$ to $\mathcal{X}'_n \times \mathcal{Y}'_n$ for sets $\mathcal{X}'_n \subseteq \mathcal{X}_n$, $|\mathcal{X}'_n| \geq \theta|\mathcal{X}_n|$, and $\mathcal{Y}'_n \subseteq \mathcal{Y}_n$, $|\mathcal{Y}'_n| \geq \theta|\mathcal{Y}_n|$, respectively.

THEOREM 5.8.10. *For a quasi-random $D$-regular graph family $\mathcal{G} := (G_n)_{n\in\mathbb{N}}$ with edge density $P$ we have*

$$\text{D}(\text{EDGE}_{G_n}) \geq \log\left(\frac{1}{P(n)}\right) - \mathcal{O}(1) \ . \tag{5.8.11}$$

*This also holds for every family of large subfunctions of* $\text{EDGE}_{\mathcal{G}}$.

PROOF. Follows from Theorem 5.8.8 and Fact 3.3.5. □

Interestingly, the right hand side looks like an entropic quantity.

This lower bound cannot be tight for worst case deterministic communication complexity, because it is actually a lower bound for the Parity-P complexity.

THEOREM 5.8.12. *For a quasi-random $D$-regular graph family $\mathcal{G} := (G_n)_{n\in\mathbb{N}}$ with edge density $P$ we have*

$$\oplus\text{P}(\text{EDGE}_{G_n}) \geq \log\left(\frac{1}{P(n)}\right) - \mathcal{O}(1) \ . \tag{5.8.13}$$

*This also holds for every family of large subfunctions of* $\text{EDGE}_{\mathcal{G}}$.

PROOF. Follows from Theorem 5.8.9 and Fact 3.3.7. □

There are a variety of constructions of sparse quasi-random graph families that have appeared in the literature. We exemplify our lower bound method with so-called Erdös-Renyi graphs that arise from finite geometries.

DEFINITION 5.8.14 (Erdös-Renyi graphs). *Let $q$ be a prime power. We define the $q$-th* Erdös-Renyi graph, *$\mathcal{ER}_q$, as follows: Let $V(\mathcal{ER}_q)$ be the points of a projective plane over $\mathbb{F}_q$. Nodes $x = (x_0, x_1, x_2)$ and $y = (y_0, y_1, y_2)$ are adjacent if $x_0y_0 + x_1y_1 + x_2y_2 = 0$ in $\mathbb{F}_q$.*

FACT 5.8.15. *The q-th Erdös-Renyi graph has* $|V(\mathcal{ER}_q)| = (q^3-1)/(q-1) = q^2+q+1$ *many nodes. It is* $D(q)$*-regular with* $D(q) := (q^2-1)/(q-1)$*, and it has the discrepancy property for* $P(q) := (q+1)/(q^2+q+1)$.

Because of $P(N(n)) = \Theta(1/N(n))$, we obtain

COROLLARY 5.8.16. $\oplus\mathrm{P}(\mathrm{EDGE}_{\mathcal{ER}_{2^n}}) \geq n - \mathcal{O}(1)$ .

We note that similar high lower bounds can be obtained for explicit families based on *Delsarte-Goethals-Turyn* graphs, generalized Erdös-Renyi graphs (defined over the projective geometry of dimension $t \geq 2$), certain incidence graphs of *generalized m-gons*, *Ramanujan graphs*, or *projective norm graphs*. See (Krivelevich & Sudakov 2006, p. 22–29) for details.

## 5.9 Concluding remarks

In this chapter we learned why Toda's Theorems are so important in communication complexity: the first one provides a complexity class, BP-Parity-P, between two alternating classes we want to separate, and this intermediate class is not based on alternation. We were able to develop a measure, namely approximate $\mathbb{F}_2$-rank, that characterizes BP-Parity-P. The tight connection between this measure and matrix rigidity over $\mathbb{F}_2$ led to concentration of measure results for the BP-Parity-P complexity. This is in contrast to previous work, where it was not clear how notions of rigidity were related to complexity classes. We were also able to prove that approximate $\mathbb{R}$-rank is a lower bound method for bounded error randomized communication complexity with a tight connection to Boolean rigidity over $\mathbb{R}$. We think that it is much easier to prove high Boolean rigidity over $\mathbb{R}$ than to prove high (classically defined) rigidity over $\mathbb{R}$, because the allowed changes are so severely restricted. In the last section we have shown that adjacency problems about sparse quasi-random graphs lead to problems with high Parity-P complexity. We think that such lower bounds also hold for the BP-Parity-P complexity.

OPEN QUESTION 5.9.1. *Let* $\mathcal{G} := (G_n)_{n\in\mathbb{N}}$ *be a quasi-random* $D$*-regular graph family with edge density* $P$*. Do we have*

$$\mathrm{BP}{\oplus}\mathrm{P}^{\mathrm{pub}}_{\epsilon}(\mathrm{EDGE}_{G_n}) \geq \log\left(\frac{1}{P(n)}\right) - \mathcal{O}(1) \ ?$$

# 6 Cover-structure graphs

## 6.1 Introduction

### 6.1.1 Cover-structure graphs

Consider a 0-1-matrix $M\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ with row set $\mathcal{X}$ and column set $\mathcal{Y}$. We call a 1-chromatic (combinatorial) rectangle *nonextendible in* $M$, if adding rows or columns to $R$ leads to a rectangle that is no longer 1-chromatic in $M$. With $M$ we associate a graph $\mathcal{G}(M)$, the *cover-structure graph (cs-graph) of* $M$. Its vertices are all nonextendible rectangles in $M$, and there is an edge between two (nonextendible) rectangles $R$ and $S$, respectively, if $R$ and $S$ have non-empty intersection, i. e. $R \cap S \neq \emptyset$. First of all, we are interested in the class **csg** of cs-graphs. Among our main findings are proofs that squares $C_4$, odd holes $C_{2n+1}$, $n \geq 2$, gem, star and watch graphs (see Figure 6.2.3) are not cs-graphs. Interestingly, even holes $C_{2n}$, $n \geq 3$, are cs-graphs. Unfortunately, we are unable to give a characterization of **csg**, but more can be said, if we look at a subclass of **csg**, namely the class **beautiful** of *beautiful graphs*. Those graphs are defined by the property that each induced subgraph is a cs-graph. By definition, **beautiful** contains square-free Berge graphs. (For a definition of the latter, see the next subsection.) The class **beautiful** is incomparable to existing classes of Berge graphs (see Table 6.3.1). We are able to show that every square-free bipartite graph is beautiful, and we are also able to characterize beautiful line graphs of square-free bipartite graphs. It turns out that the latter are just *Path-or-Even-Cycle-of-Cliques* graphs (see Definition 6.3.8).

### 6.1.2 Perfect graphs

Shannon (1948, 1965) considered zero-error data transmission and reduced the problem of determining the zero-error channel capacity to a problem in graph theory, namely calculating $\sup_{n\to\infty} \frac{1}{n} \log \omega(G^n)$ (now called *Shannon zero-error capacity*), where $G$ is a graph associated with the given channel, $G^n$ is its $n$-th graph power, and $\omega(G)$ is the clique number of $G$. The $n$-th graph power $G^n$ is the strong graph product of $n$ copies of $G$; given graphs $G_1$ and $G_2$ the strong graph product is a graph with vertex set $V(G_1) \times V(G_2)$ and two distinct vertices are connected iff they are adjacent or equal in each coordinate. Determining the Shannon zero-error capacity is extremely hard in general, e. g. see Alon & Lubetzky (2006); Lovász (1979), but easily solved for so called *perfect graphs*, introduced in Berge (1961). These are graphs for which the chromatic and clique number have the same value for each induced subgraph. For an excellent introduction to the theory of perfect graphs we refer the reader to Ramírez-Alfonsín & Reed (2001). Berge conjectured that a graph is perfect iff it does not contain any odd holes or odd antiholes. An induced cycle of odd length at least 5 is called an *odd hole*, while an induced subgraph that is the complement of an odd hole is called an *odd antihole*. Graphs without odd holes and odd antiholes are called *Berge graphs*. The above conjecture was known as the *Strong Perfect Graph Conjecture*, which, based on a series of works, especially Conforti, Cornuéjols & Vušković (2004), was finally answered in the affirmative by Chudnovsky, Robertson, Seymour & Thomas (2006). Our characterizations of beautiful bipartite graphs and beautiful line graphs of bipartite graphs are motivated by the decomposition theorem of square-free Berge graphs presented in Conforti *et al.* (2004).

### 6.1.3 A problem in communication complexity

We study cs-graphs, because their definition is motivated by a problem in communication complexity, namely the $C^D$-vs.-$C^P$ problem, described in Section 3.2, a challenging and long-standing open problem. A possible strategy to solve this problem might be to start with an arbitrary minimal (not necessarily protocol induced) partition of the input space with monochromatic rectangles, and then to cut some of those rectangles into smaller ones until we finally arrive at a partition that is protocol induced. Because we want to keep the size of the new partition as small as possible, i. e. close to the size of the one we started with, we want to cut as few rectangles as possible. Thus, it might be useful to have information about the *relative positions* of the rectangles. This is where cs-graphs come into play. We embed the rectangles of the partition into nonextendible ones. If we know that certain rectangle configurations are not possible, we might be able to prove that the cuts we make are not severe, i. e. they do not create many new rectangles. It is useful to have a characterization of beautiful graphs, because then it might be possible to apply the following variant of the above strategy: we cut rectangles of the $C^D$-partition until we arrive at rectangle sets that are not protocol induced yet, but each of them induces a beautiful cs-graph, and thus can be handled with only few more cuts, because beautiful graphs seem to have a much simpler structure than general cs-graphs.
Of course, we do not know if the strategies described above can be realized, and thus we do not know if the study of beautiful and cs-graphs will lead to this application in communication complexity. But to the author's knowledge there does not exist any alternative strategy in the published literature to tackle the $C^D$-vs.-$C^P$-problem.

## 6.2 Cover-structure graphs

### 6.2.1 Definition and easy observations

In this subsection we define cover-structure graphs, prove several easy results about them, and state some of their structural properties.

Definition 6.2.1 (Nonextendible rectangle). *Let $M$ be a function or relation matrix over $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$. A rectangle $R$ is* nonextendible *iff $R$ is monochromatic in $M$ and adding rows or columns to $R$ results in a non-monochromatic rectangle.*

Definition 6.2.2 (Cover-structure graph). *Let $M$ be a function or relation matrix over $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$.*

- *We associate with $M$ its* cover-structure graph $\mathcal{G}(M) := (\mathcal{V}(M), \mathcal{E}(M))$ *(*cs-graph *for short), where*

$$\mathcal{V}(M) := \{R \mid R \text{ nonextendible rectangle in } M\} \ ,$$
$$\mathcal{E}(M) := \{\{R, R'\} \mid R, R' \in \mathcal{V}(M), R \neq R', R \cap R' \neq \emptyset\} \ .$$

- *Let $z \in \mathcal{Z}$. We also associate with $M$ its $z$-*chromatic cover-structure graph *$\mathcal{G}^z(M) := (\mathcal{V}^z(M), \mathcal{E}^z(M))$, where*

$$\mathcal{V}^z(M) := \{R \mid R \text{ nonextendible } z\text{-chromatic rectangle in } M\} \ ,$$
$$\mathcal{E}^z(M) := \{\{R, R'\} \mid R, R' \in \mathcal{V}^z(M), R \neq R', R \cap R' \neq \emptyset\} \ .$$

The following result might lead to the conclusion that cs-graphs are uninteresting. However, for function matrices the situation is completely different, as we will see in Theorem 6.2.6.

THEOREM 6.2.3. *Let $G$ be an arbitrary graph. Then there exists a* relation *matrix $M$ such that $G =_{\text{iso}} \mathcal{G}(M)$.*

PROOF. W. l. o. g. assume $G = ([n], E)$. We define the $1 \times n^2$-block matrix $M$ with values in $\mathcal{P}([n])$ by $M := (B^{(1)}, \ldots, B^{(n)})$, where each block $B^{(i)}$ is a $1 \times n$-matrix defined by $B^{(i)}_{1,j} := \{i, j\}$, if $\{i, j\} \in E$, and $B^{(i)}_{1,j} := \{i\}$ otherwise. For each color $i \in [n]$ there exists exactly one nonextendible rectangle $R_i := \{1\} \times \{j \mid i \in M_{1,j}\}$. Thus, $\mathcal{V}(M) = \{R_i \mid i \in [n]\}$. If $\{i, j\} \in E$, then $R_i$ and $R_j$ intersect in block position $B^{(i)}_{1,j}$ (and in $B^{(j)}_{1,i}$) implying $\{R_i, R_j\} \in \mathcal{E}(M)$. Conversely, if $\{R_i, R_j\} \in \mathcal{E}(M)$, then there exist indices $k, l \in [n]$ such that $R_i$ and $R_j$ intersect in $B^{(k)}_{1,l}$. The case $k \notin \{i, j\}$ cannot occur by construction ($|B^{(i)}_{1,j}| \leq 2$). W. l. o. g. assume $k = i$. Necessarily, $B^{(i)}_{1,l} = \{i, j\}$. Thus, $l = j$ and $\{i, j\} \in E$. We conclude $\mathcal{E}(M) = \{\{R_i, R_j\} \mid \{i, j\} \in E\}$ proving $G =_{\text{iso}} \mathcal{G}(M)$. □

Given $z \in \mathcal{Z}$ and a function matrix $M$ over $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$, define the corresponding 0-1-matrix $M^{(z)}$ by $M^{(z)}_{x,y} := [M_{x,y} = z]$. As monochromatic rectangles with different colors do not intersect for function matrices, we get $\mathcal{G}(M) =_{\text{iso}} \biguplus_{z \in \mathcal{Z}} \mathcal{G}^z(M^{(z)})$. Thus, we only need to deal with cs-graphs of function matrices over finite sets $\mathcal{X}, \mathcal{Y}$ and $\mathcal{Z} = \{0, 1\}$. From here on, when we talk about *matrices*, we mean function matrices over finite sets $\mathcal{X}, \mathcal{Y}$ and $\mathcal{Z} = \{0, 1\}$. We also write $\mathcal{G}(M)$, when we mean $\mathcal{G}^1(M)$. We call matrices $M$ with $G =_{\text{iso}} \mathcal{G}(M)$ *representations* of $G$. We denote the *class of cs-graphs*, i. e. the class of graphs which can be represented by function matrices, with **csg**.

The independent set $\overline{K}_n$ and the complete graph $K_n$, both defined on $n$ nodes, are cs-graphs, as can be seen by looking at the identity matrix $E_n$ and the triangular matrix $T_n$, respectively, Here, $T_n$ is an $n \times n$-matrix with ones on and above the diagonal and zeros below. Even cycles $C_{2n}$, $n \in \mathbb{N}$, are cs-graphs, too. This is obvious for $n \leq 2$. In case $n \geq 3$ we have $C_{2n} =_{\text{iso}} \mathcal{G}({}_{\text{rep}}\mathrm{C}_{2n})$, where the representation ${}_{\text{rep}}\mathrm{C}_{2n}$ is the $n \times n$-matrix shown in Figure 6.2.1.

In the sequel we only consider connected cs-graphs. On the one hand, if $G$ has connected components $G_1$ and $G_2$ represented by $M_1$ and $M_2$, respectively, then the block diagonal matrix $M := \operatorname{diag}(M_1, M_2)$ is a representation of $G$. On the other hand, one can show that if $G$ is a cs-graph, then its components $G_1$ and $G_2$ are also cs-graphs: If in a representation $M$ of $G$ a rectangle $R_1$ representing a node $v_1 \in G_1$ would share a row or column with a rectangle $R_2$ representing a node $v_2 \in G_2$, then there would exist a nonextendible rectangle $J$ that has non-empty intersection with both $R_1$ and $R_2$. Thus, the rectangle $J$ would represent a node $v$ in $G$ adjacent to both $v_1$ and $v_2$ in contradiction to the assumption that $G_1$ and $G_2$ are different connected components. If $A_i$ and $B_i$ denote the rows and columns covered by rectangles representing nodes in $G_i$, then $A_1 \cap A_2, B_1 \cap B_2 = \emptyset$. Thus, a permutation of the rows and columns of $M$ yields a representation $\operatorname{diag}(M_1, M_2)$ of $G$.

## 6.2.2 Graphs that are not cs-graphs

In this subsection we show that in contrast to the case of relation matrices not every graph is a cs-graph of a function matrix. An important observation is that nonextendible rectangles cannot intersect in an arbitrary fashion. Only two modes of intersection are possible, namely *cross* and *spade* situations (see Figure 6.2.2).

DEFINITION 6.2.4 (Cross and spade situation). *Let $M$ be a matrix, and let $R_i := A_i \times B_i \in \mathcal{V}(M)$, $i \in [2]$, such that $\{R_1, R_2\} \in \mathcal{E}(M)$.*

- *If $A_1 \subsetneq A_2$ and $B_2 \subsetneq B_1$, then we have a* cross situation *cross$(R_1, R_2)$.*

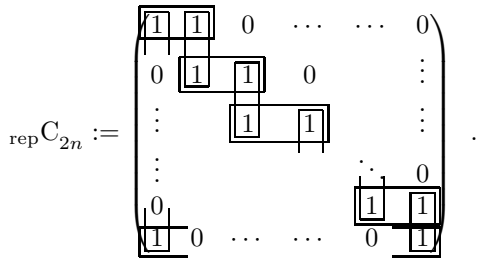

Figure 6.2.1: Representation of an even cycle $C_{2n}$, $n \geq 3$

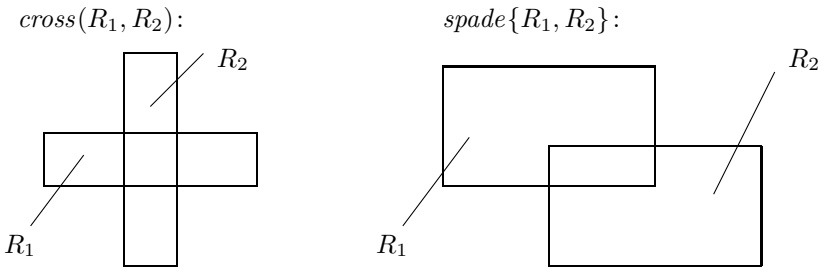


Figure 6.2.2: Cross and spade situations

- *If* $A_1 - A_2, A_2 - A_1, B_1 - B_2, B_2 - B_1 \neq \emptyset$, *then we have a* spade situation $spade\{R_1, R_2\}$.

Note that while $spade\{R_1, R_2\}$ implies $spade\{R_2, R_1\}$ in case $cross(R_1, R_2)$ the situation $cross(R_2, R_1)$ does not occur. In case we do not care which cross situation holds, we let $cross\{R_1, R_2\} := cross(R_1, R_2) \vee cross(R_2, R_1)$ denote the symmetrized version.

In the following lemma we list helpful observations which we will extensively use in the sequel.

LEMMA 6.2.5 (Proof tools). *Let $M$ be a matrix, and let $R_i := A_i \times B_i \in \mathcal{V}(M)$, $i \in [3]$, be arbitrary nonextendible rectangles.*

(i) *If $\{R_1, R_2\} \in \mathcal{E}(M)$, then exactly one of the following situations occurs:*

$$cross(R_1, R_2) \ , \ cross(R_2, R_1) \ , \ spade\{R_1, R_2\} \ .$$

(ii) *$cross(R_1, R_2)$ and $cross(R_2, R_3)$ implies $cross(R_1, R_3)$.*
*Especially, we have $\{R_1, R_3\} \in \mathcal{E}(M)$ in this case.*

(iii) *$spade\{R_1, R_2\}$ implies $K_4 \leq_{\text{iso}} \mathcal{G}(M)$.*

(iv) *If $cross(R_1, R_2)$, $cross(R_3, R_2)$, $\{R_1, R_3\} \notin \mathcal{E}(M)$, and $B_2 \subsetneq B_1 \cap B_3$, then there exists $R_4 \in \mathcal{V}(M)$ such that $\{R_i, R_4\} \in \mathcal{E}(M)$ for all $i \in [3]$.*

PROOF. (i) By case distinction: *Case* $A_1 = A_2$: Here $R_1 = R_2$, or at least one of $R_1, R_2$ is extendible, a contradiction. *Case* $A_1 \subsetneq A_2$: If $B_1 \subsetneq B_2$ or $B_1 - B_2, B_2 - B_1 \neq \emptyset$ then $R_1$ is extendible, a contradiction. If $B_2 \subsetneq B_1$ then we have $cross(R_1, R_2)$. *Case* $A_2 \subsetneq A_1$: Analogous to $A_1 \subsetneq A_2$. *Case* $A_1 - A_2, A_2 - A_1 \neq \emptyset$: All cases for $B_1$ and $B_2$ are analogous to the previous ones, except $B_1 - B_2, B_2 - B_1 \neq \emptyset$, where we have a spade situation $spade\{R_1, R_2\}$.

(ii) From $cross(R_1, R_2)$ and $cross(R_2, R_3)$ it follows $A_1 \subsetneq A_2$, $B_2 \subsetneq B_1$ and $A_2 \subsetneq A_3$, $B_3 \subsetneq B_2$, respectively. Thus, $A_1 \subsetneq A_3$ and $B_3 \subsetneq B_1$, which implies $cross(R_1, R_3)$.

(iii) Let $S_3$ and $S_4$ be arbitrary nonextendible rectangles in $M$ such that $S_3$ covers $(A_1 \cap A_2) \times (B_1 \cup B_2)$, and $S_4$ covers $(A_1 \cup A_2) \times (B_1 \cap B_2)$, respectively. Clearly, the rectangles $R_1, R_2, S_3, S_4$ are pairwise distinct. They intersect pairwise, because all of them cover $(A_1 \cap A_2) \times (B_1 \cap B_2)$ . Thus, we have $\mathcal{G}(M)(\{R_1, R_2, S_3, S_4\}) =_{\text{iso}} K_4$.

(iv) Let $R_4 \in \mathcal{V}(M)$ be an arbitrary nonextendible rectangle covering $(A_1 \cup A_3) \times (B_1 \cap B_3)$. From $cross(R_1, R_2)$ it follows $A_1 \subsetneq A_2$. As $B_2 \subsetneq B_1 \cap B_3$ by assumption, we get $R_4 \cap R_2 \neq \emptyset$ and $R_4 \neq R_2$. By construction we also have $R_4 \cap R_1, R_4 \cap R_3 \neq \emptyset$. From $\{R_1, R_3\} \notin \mathcal{E}(M)$ and $\emptyset \neq B_2 \subsetneq B_1 \cap B_3$ we derive $A_1 \cap A_3 = \emptyset$. Thus, $R_4 \neq R_1$ and $R_4 \neq R_3$. □

Now we can show that not all graphs are cs-graphs:

THEOREM 6.2.6. *The square $C_4$, odd holes $C_{2n+1}$, $n \geq 2$, and the graphs gem, watch and star[1] (see Figure 6.2.3) are not cs-graphs.*

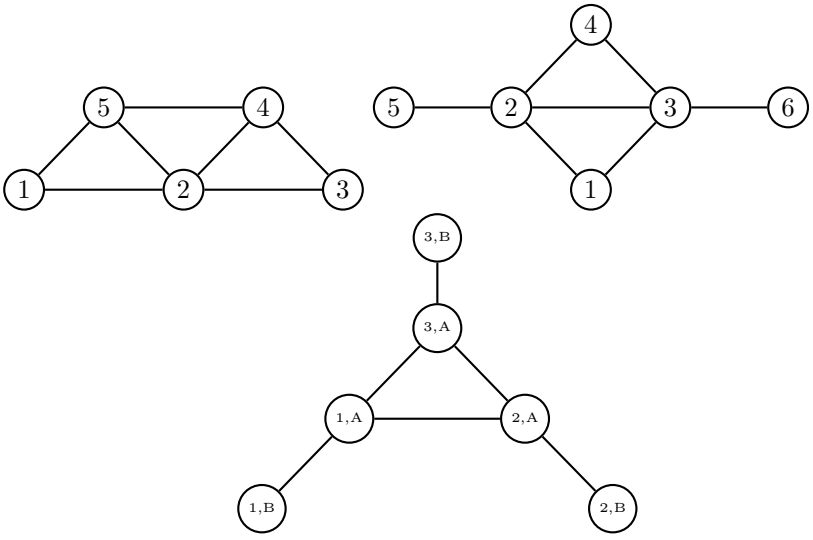


Figure 6.2.3: Gem, watch and star

PROOF. Due to the many case distinctions we recommend that the reader visualizes the proofs by drawing the cross situations under consideration.

We assume for a contradiction that $C_4$ is a cs-graph. Then there exists a matrix $M$ such that $C_4 =_{\text{iso}} \mathcal{G}(M)$. We have $\mathcal{V}(M) = \{R_1, \ldots, R_4\}$ and $\mathcal{E}(M) = \{\{R_1, R_2\}, \{R_2, R_3\}, \{R_3, R_4\}, \{R_4, R_1\}\}$. By Lemma 6.2.5(i) and (iii) for $R_i, R_{i+1}$ and $R_4, R_1$ only cross situations are possible, because $K_4 \not\leq_{\text{iso}} C_4$. W. l. o. g. we assume $cross(R_1, R_2)$. Then by Lemma 6.2.5(ii) we must have $cross(R_3, R_2)$, as $C_3 \not\leq_{\text{iso}} C_4$. Applying Lemma 6.2.5(iv) yields $B_1 \cap B_3 = B_2$. An analogous argumentation (consider the

[1]The star graph is also called *net* in many publications.

transpose of $M$) for $R_2, R_3, R_4$ yields $A_2 \cap A_4 = A_3$. From $R_1 \cap R_3 = \emptyset$ and $B_1 \cap B_3 = B_2 \neq \emptyset$ it follows $A_1 \cap A_3 = \emptyset$. Then we have $A_4 = A_3 \cup (A_4 - A_2)$, and thus $A_4 \cap A_1 = (A_3 \cap A_1) \cup ((A_4 - A_2) \cap A_1) = \emptyset \cup \emptyset = \emptyset$ using $A_1 \subseteq A_2$. But this implies $R_1 \cap R_4 = \emptyset$ contradicting $\{R_1, R_4\} \in \mathcal{E}(M)$. We conclude that $C_4$ cannot be a cs-graph.

We assume for a contradiction that $C_{2n+1}$ is a cs-graph for $n \geq 2$. Then there exists a matrix $M$ such that $C_{2n+1} =_{\text{iso}} \mathcal{G}(M)$. We have $\mathcal{V}(M) = \{R_1, \ldots, R_{2n+1}\}$ and $\mathcal{E}(M) = \{\{R_i, R_{i+1}\} \mid i \in [2n]\} \cup \{\{R_{2n+1}, R_1\}\}$. Only cross situations are possible, because of $K_4 \not\leq_{\text{iso}} C_{2n+1}$ and Lemma 6.2.5(i) and (iii). W. l. o. g. we assume $\mathit{cross}(R_1, R_2)$. As $C_3 \not\leq_{\text{iso}} C_{2n+1}$ iteratively applying Lemma 6.2.5(ii) yields the sequence $\mathit{cross}(R_3, R_2)$, $\mathit{cross}(R_3, R_4)$, ..., $\mathit{cross}(R_{2n+1}, R_{2n})$, and thus $\mathit{cross}(R_{2n+1}, R_1)$. But going backwards starting from $\mathit{cross}(R_1, R_2)$ gives us $\mathit{cross}(R_1, R_{2n+1})$. We get $\mathit{cross}(R_{2n+1}, R_1)$ and $\mathit{cross}(R_1, R_{2n+1})$, a contradiction. We conclude that $C_{2n+1}$ cannot be a cs-graph.

We assume for a contradiction that gem is a cs-graph, i. e. gem $=_{\text{iso}} \mathcal{G}(M)$ for a matrix $M$. We have $\mathcal{V}(M) = \{R_1, \ldots, R_5\}$ and $\mathcal{E}(M) = \{\{R_1, R_2\}, \{R_1, R_5\}, \{R_2, R_3\}, \{R_2, R_4\}, \{R_2, R_5\}, \{R_3, R_4\}, \{R_4, R_5\}\}$. As $K_4 \not\leq_{\text{iso}}$ gem, only cross situations are possible by Lemma 6.2.5(i) and (iii). W. l. o. g. we assume $\mathit{cross}(R_1, R_2)$. We have $\{R_1, R_3\}, \{R_1, R_4\} \notin \mathcal{E}(M)$ implying $\mathit{cross}(R_3, R_2)$ and $\mathit{cross}(R_4, R_2)$, respectively.

1. Assume $\mathit{cross}(R_3, R_4)$.
   - 1.1. Assume $\mathit{cross}(R_1, R_5)$. $\mathit{cross}(R_2, R_5)$ implies $\mathit{cross}(R_3, R_5)$ contradicting $R_3 \cap R_5 = \emptyset$. Thus, assume $\mathit{cross}(R_5, R_2)$.
     - 1.1.1. Assume $\mathit{cross}(R_4, R_5)$. Then $A(R_3) \subseteq A(R_4) \subseteq A(R_5)$ and $B(R_5) \subseteq B(R_4) \subseteq B(R_3)$. But $R_3 \cap R_5 = (A(R_3) \cap A(R_5)) \times (B(R_3) \cap B(R_5)) \supseteq A(R_3) \times B(R_5) \neq \emptyset$, a contradiction.
     - 1.1.2. Assume $\mathit{cross}(R_5, R_4)$. We must have $A(R_1) \cap A(R_4) = \emptyset$, as $B(R_1) \cap B(R_4) \supseteq B(R_2) \neq \emptyset$ and $R_1 \cap R_4 = \emptyset$. But then $A(R_5) \subseteq A(R_4)$ implies $A(R_1) \cap A(R_5) = \emptyset$ contradicting $R_1 \cap R_5 \neq \emptyset$.
   - 1.2. Assume $\mathit{cross}(R_5, R_1)$. We still have $A(R_1) \cap A(R_4) = \emptyset$. But $A(R_5) \subseteq A(R_1)$ implies $A(R_4) \cap A(R_5) = \emptyset$ contradicting $R_4 \cap R_5 \neq \emptyset$.

2. Assume $\mathit{cross}(R_4, R_3)$. $\mathit{cross}(R_5, R_4)$ implies $\mathit{cross}(R_5, R_3)$ contradicting $R_3 \cap R_5 = \emptyset$. Thus, assume $\mathit{cross}(R_4, R_5)$.
   - 2.1. If $\mathit{cross}(R_2, R_5)$ then $\mathit{cross}(R_3, R_5)$ contradicting $R_3 \cap R_5 = \emptyset$.
   - 2.2. If $\mathit{cross}(R_5, R_2)$ then $\emptyset \neq B(R_2) \subseteq B(R_3) \cap B(R_5)$. As $R_3 \cap R_5 = \emptyset$, it must hold $A(R_3) \cap A(R_5) = \emptyset$. But $A(R_4) \subseteq A(R_5)$. We finally get $A(R_3) \cap A(R_4) = \emptyset$ contradicting $R_3 \cap R_4 \neq \emptyset$.

We conclude that gem cannot be a cs-graph.

We assume for a contradiction that star is a cs-graph, i. e. star $=_{\text{iso}} \mathcal{G}(M)$ for a matrix $M$. We have $\mathcal{V}(M) = \{R_{i,A}, R_{i,B} \mid i \in [3]\}$ and $\mathcal{E}(M) = \{\{R_{i,A}, R_{i,B}\} \mid i \in [3]\} \cup \{\{R_{1,A}, R_{2,A}\}, \{R_{2,A}, R_{3,A}\}, \{R_{1,A}, R_{3,A}\}\}$. As $K_4 \not\leq_{\text{iso}}$ star, only cross situations are possible by Lemma 6.2.5(i) and (iii). W. l. o. g. we assume $\mathit{cross}(R_{1,A}, R_{2,A})$.

1. Assume $\mathit{cross}(R_{2,A}, R_{3,A})$. $\mathit{cross}(R_{2,A}, R_{2,B})$ implies $\mathit{cross}(R_{1,A}, R_{2,B})$ contradicting $R_{1,A} \cap R_{2,B} = \emptyset$. $\mathit{cross}(R_{2,B}, R_{2,A})$ implies $\mathit{cross}(R_{2,B}, R_{3,A})$ contradicting $R_{2,B} \cap R_{3,A} = \emptyset$.

2. Assume $\mathit{cross}(R_{3,A}, R_{2,A})$.
   - 2.1. Assume $\mathit{cross}(R_{1,A}, R_{3,A})$. $\mathit{cross}(R_{3,A}, R_{3,B})$ implies $\mathit{cross}(R_{1,A}, R_{3,B})$ contradicting $R_{1,A} \cap R_{3,B} = \emptyset$. $\mathit{cross}(R_{3,B}, R_{3,A})$ implies $\mathit{cross}(R_{3,B}, R_{2,A})$ contradicting $R_{2,A} \cap R_{3,B} = \emptyset$.
   - 2.2. Assume $\mathit{cross}(R_{3,A}, R_{1,A})$. $\mathit{cross}(R_{1,A}, R_{1,B})$ implies $\mathit{cross}(R_{3,A}, R_{1,B})$ contradicting $R_{3,A} \cap R_{1,B} = \emptyset$. $\mathit{cross}(R_{1,B}, R_{1,A})$ implies $\mathit{cross}(R_{1,B}, R_{2,A})$ contradicting $R_{2,A} \cap R_{1,B} = \emptyset$.

We conclude that star cannot be a cs-graph.

We assume for a contradiction that watch is a cs-graph, i. e. watch $=_{\text{iso}} \mathcal{G}(M)$ for a matrix $M$. We have $\mathcal{V}(M) = \{R_1, \ldots, R_6\}$ and $\mathcal{E}(M) = \{\{R_5, R_2\}, \{R_2, R_1\}, \{R_2, R_3\}, \{R_2, R_4\}, \{R_1, R_3\}, \{R_4, R_3\}, \{R_3, R_6\}\}$. As watch is $K_4$-free, only cross situations are possible by Lemma 6.2.5(i) and (iii). W. l. o. g. we assume $cross(R_1, R_2)$.

1. Assume $cross(R_2, R_3)$. $cross(R_2, R_5)$ implies $cross(R_1, R_5)$ contradicting $R_1 \cap R_5 = \emptyset$. $cross(R_5, R_2)$ implies $cross(R_5, R_3)$ contradicting $R_3 \cap R_5 = \emptyset$.
2. Assume $cross(R_3, R_2)$.
   2.1. Assume $cross(R_1, R_3)$. $cross(R_3, R_6)$ implies $cross(R_1, R_6)$ contradicting $R_1 \cap R_6 = \emptyset$. $cross(R_6, R_3)$ implies $cross(R_6, R_2)$ contradicting $R_2 \cap R_6 = \emptyset$.
   2.2. Assume $cross(R_3, R_1)$. $cross(R_2, R_4)$ implies $cross(R_1, R_4)$ contradicting $R_1 \cap R_4 = \emptyset$. Thus, assume $cross(R_4, R_2)$. If $cross(R_3, R_4)$ then $R_2 \cap R_3 \subseteq R_4$ implying $R_1 \cap R_4 \neq \emptyset$, a contradiction. $cross(R_4, R_3)$ implies $cross(R_4, R_1)$, but then we have $R_1 \cap R_4 \neq \emptyset$, a contradiction.

We conclude that watch cannot be a cs-graph. □

## 6.3 Beautiful graphs

We have seen in the last section that **csg** does not contain all graphs. As squares and odd holes are "forbidden", the previous results motivate the following definition:

Definition 6.3.1. *A graph is* beautiful *iff every induced subgraph is a cs-graph.*

We denote with **beautiful** the class of beautiful graphs. Clearly, from Theorem 6.2.6 we obtain:

Theorem 6.3.2. *Every beautiful graph is a square-free Berge graph.* □

The opposite is not true, as e. g. a star is square-free and Berge, but not beautiful. A comparison with known classes of perfect/Berge graphs (see e. g. Hougardy (2006); McKee & McMorris (1999) and Table 6.3.1 below comparing cs-graphs, $K_4$-free cs-graphs and the class of beautiful graphs with known classes of square-free perfect graphs, namely interval, split, threshold, triangulated and trivially perfect graphs) yields that **beautiful** is a new class of Berge graphs. In Table 6.3.1 we list the interesting class of $K_4$-free cs-graphs, because such graphs cannot be represented by matrices containing spade situations. We conjecture that this class coincides with the class of $K_4$-free beautiful graphs. We state without proof[2] that the list of forbidden induced subgraphs of beautiful graphs in Theorem 6.2.6 is complete up to connected graphs of order $n \leq 7$.

We explore the structure of beautiful graphs in the spirit of Conforti, Cornuéjols & Vušković. Recall their decomposition theorem about square-free perfect graphs:

Theorem 6.3.3 (Conforti *et al.* 2004). *A square-free perfect graph is bipartite or the line graph of a bipartite graph or has a star cutset or a 2-join.*

We leave the notions "star cutset" and "2-join" undefined here, because we do not need them in the sequel.

We are able to give characterizations of beautiful square-free bipartite graphs (Subsection 6.3.1) and beautiful line graphs of square-free bipartite graphs (Subsection 6.3.2).

[2] The respective proof that we have not included here due to its disproportionate length contains a list of about a thousand graphs such that for each graph a representation is given or a statement that it contains one of the forbidden induced subgraphs.

| | **interval** | **split** | **threshold** | **triangulated** | **triv.perfect** |
|---|---|---|---|---|---|
| **beautiful** | $\not\supseteq$, gem<br>$\not\subseteq$, $C_6$ | $\not\supseteq$, star<br>$\not\subseteq$, $\overline{C}_4$ | $\not\supseteq$, gem<br>$\not\subseteq$, $\overline{C}_4$ | $\not\supseteq$, gem<br>$\not\subseteq$, $C_6$ | $\not\supseteq$, star<br>$\not\subseteq$, $P_4$ |
| $K_4-$**free**<br>**csg** | $\not\supseteq$, gem<br>$\not\subseteq$, $C_6$ | $\not\supseteq$, star<br>$\not\subseteq$, $\overline{C}_4$ | $\not\supseteq$, gem<br>$\not\subseteq$, $\overline{C}_4$ | $\not\supseteq$, gem<br>$\not\subseteq$, $C_6$ | $\not\supseteq$, star<br>$\not\subseteq$, $P_4$ |
| **csg** | $\not\supseteq$, gem<br>$\not\subseteq$, $C_6$ | $\not\supseteq$, star<br>$\not\subseteq$, $\overline{C}_4$ | $\not\supseteq$, gem<br>$\not\subseteq$, $\overline{C}_4$ | $\not\supseteq$, gem<br>$\not\subseteq$, $C_6$ | $\not\supseteq$, star<br>$\not\subseteq$, $P_4$ |

Table 6.3.1: Comparisons of graph classes

### 6.3.1 All square-free bipartite graphs are beautiful

In this subsection we show that all square-free bipartite graphs are beautiful.

PROPOSITION 6.3.4. *Every square-free bipartite graph is a cs-graph.*

PROOF. Let $G := (U \cup V, E)$ be square-free and bipartite. W. l. o. g. assume $U = [m]$ and $V = [n]$. Define the adjacency matrix of $G$ as the $m \times n$-matrix $I$, where $I_{u,v} := [\{u, v\} \in E]$, $u \in U$, $v \in V$. Let

$$M := \begin{pmatrix} I & E_m \\ E_n & 0 \end{pmatrix} .$$

Consider any $R = A \times B \in \mathcal{V}(M)$. If $R$ covers elements in $E_m$, then necessarily $|A| = 1$. Thus, there exists $u \in [m]$ such that $A = \{u\}$ and $B = \{v \in [n] \mid \{u, v\} \in E\} \cup \{n + u\}$. If $R$ covers elements in $E_n$, then necessarily $|B| = 1$. Thus, there exists $v \in [n]$ such that $B = \{v\}$ and $A = \{u \in [m] \mid \{u, v\} \in E\} \cup \{m + v\}$. Suppose $R$ covers only elements in $I$. Then necessarily, $|A|, |B| \geq 2$. Then there exist distinct $u_1, u_2 \in A \subseteq [m]$ and distinct $v_1, v_2 \in B \subseteq [n]$ such that $I_{u_i,v_j} = 1$, $i \in [2]$, $j \in [2]$. This means $\{u_1, v_1\}, \{v_1, u_2\}, \{u_2, v_2\}, \{v_2, u_1\} \in E$. As $G$ is bipartite, we have $\{u_1, u_2\}, \{v_1, v_2\} \notin E$. Thus, $C_4 \leq_{\text{iso}} G$, a contradiction. We conclude $G =_{\text{iso}} \mathcal{G}(M)$. □

As every induced subgraph of a square-free bipartite graph is square-free bipartite, from Proposition 6.3.4 we immediately obtain:

THEOREM 6.3.5. *Every square-free bipartite graph is beautiful.* □

### 6.3.2 Characterization of beautiful line graphs of square-free bipartite graphs

In this subsection we completely describe beautiful line graphs of square-free bipartite graphs. Here, the situation is more complicated.

We begin by fixing some notation. In this subsection we let $\tilde{G} := (U^l \cup U^r, \tilde{E})$ be a square-free bipartite graph, and we let $G := L(\tilde{G}) = (V, E)$, $V := \tilde{E}$, be its line graph. We may assume that $U^l$ and $U^r$ do not contain any isolated nodes in $\tilde{G}$, and that $G$ is connected. For $u \in U^l$ define $K^l_u := \{e \in V \mid u \in e\}$. The set $K^r_v$ is defined analogously for $v \in U^r$. Each $K^l_u$ is a clique in $G$ and $\{K^l_u \mid u \in U^l\}$ is a partition of $V$, the *left clique partition of* $G$. The *right clique partition* is defined analogously. We prove all results for the *left side* only, but of course, they also hold for the *right side*. We need the following claim:

CLAIM 6.3.6. *Let $u, u' \in U^l$, $u \neq u'$, be arbitrary. Then between $K^l_u$ and $K^l_{u'}$ there is at most one edge, i. e. $\left|E(K^l_u, K^l_{u'})\right| \leq 1$.*

PROOF. We assume the opposite for a contradiction. Let $e_1, e_2 \in K^l_u$ be distinct elements. We distinguish two cases:

1. There exists $d \in K^l_{u'}$ such that $\{e_1, d\}, \{e_2, d\} \in E$. Then there exist distinct $v_1, v_2 \in U^r$ such that $e_i = \{u, v_i\}$, $i \in [2]$. As $\{e_1, d\} \in E$, we obtain $d = \{u, v_1\}$, and also $d = \{u, v_2\}$ by $\{e_2, d\} \in E$, a contradiction.

2. There exist distinct $d_1, d_2 \in K^l_{u'}$ such that $\{e_1, d_1\}, \{e_2, d_2\} \in E$. By the argument above, we have $\{e_1, d_2\}, \{e_2, d_1\} \notin E$. As $\{e_1, e_2\}, \{d_1, d_2\} \in E$ we get $C_4 \leq_{\text{iso}} G$, again a contradiction.

We conclude that there is at most one edge between $K^l_u$ and $K^l_{u'}$. □

For $u \in U^l$ define the set of *connection nodes* as

$$B^l_u := \{e \in K^l_u \mid \exists u' \in U^l \colon u \neq u', e \text{ adjacent to } K^l_{u'}\} \ .$$

We call a clique $K^l_u$ *non-trivial*, if $|K^l_u| \geq 2$, and *trivial* otherwise. In addition, we collect the names of non-trivial left cliques in the set

$$F^l := \{u \in U^l \mid K^l_u \text{ non-trivial}\} \ .$$

LEMMA 6.3.7. *Let $G$ be a graph as defined at the beginning of this section. Assume that $G$ is beautiful. Then the following statements hold:*

(i) *Assume there exist distinct $u, u' \in U^l$, distinct $e_1, e_2 \in K^l_u$, and $d \in K^l_{u'}$ such that $\{d, e_1\} \in E$. Let $G =_{\text{iso}} \mathcal{G}(M)$ for a matrix $M$. If $R(v)$ denotes the nonextendible rectangle corresponding to $v \in V(G)$ in $M$, then we must have $\mathit{cross}\{R(e_1), R(e_2)\}$ and $\mathit{cross}\{R(e_1), R(d)\}$.*

(ii) *In each left clique there exist at most two nodes adjacent to other left cliques. Especially, we must have $|B^l_u| \leq 2$ for each $u \in U^l$.*

(iii) *Let $u_i \in U^l$ be pairwise distinct, and let $e_i \in K^l_{u_i}$, $i \in [3]$. If the set of nodes $\{e_i \mid i \in [3]\}$ forms a triangle in $G$, then at least one of the cliques $K^l_{u_i}$ is trivial.*

(iv) *$\mathcal{G}(\bigcup_{u \in F^l} B^l_u)$ is bipartite.*

PROOF. (i) We assume for a contradiction that we have $\mathit{spade}\{R(e_1), R(e_2)\}$. By Lemma 6.2.5(iii) there exist distinct $g_1, g_2 \in V$ such that $\{e_1, e_2, g_1, g_2\}$ is a $K_4$ in $G$. By Claim 6.3.6 we get $g_1, g_2 \in K^l_u$. We distinguish two cases:

1. In case $\mathit{cross}\{R(e_1), R(d)\}$ we must have $\{d, e_1\}, \{d, g_1\} \in E$ or $\{d, e_1\}, \{d, g_2\} \in E$, which is impossible by Claim 6.3.6.

2. In case $\mathit{spade}\{R(e_1), R(d)\}$ by Lemma 6.2.5(iii) there exist distinct $h_1, h_2 \in V$ such that $\{e_1, d, h_1, h_2\}$ is a $K_4$ in $G$. In addition, the nodes $g_1, g_2, h_1, h_2$ are pairwise distinct. If $h_1 \in K^l_u$, then $\{d, e_1\}, \{d, h_1\} \in E$ contradicting Claim 6.3.6. If $h_1 \notin K^l_u$, then there exists $u'' \in U^l$, $u \neq u''$, such that $h_1 \in K^l_{u''}$. But then $\{h_1, e_1\}, \{h_1, g_1\} \in E$, again contradicting Claim 6.3.6.

We conclude that the situation $\mathit{spade}\{R(e_1), R(e_2)\}$ cannot occur. By Lemma 6.2.5(i) we obtain $\mathit{cross}\{R(e_1), R(e_2)\}$ proving the first statement in (i).

For the second statement in (i), we assume for a contradiction that we have the situation $\mathit{spade}\{R(e_1), R(d)\}$. By Lemma 6.2.5(iii) there exist distinct $g_1, g_2 \in V$ such that $\{e_1, d, g_1, g_2\}$ is a $K_4$ in $G$. By Claim 6.3.6 there must exist $u_1, u_2 \in U^l$, $u, u', u_1, u_2$ pairwise distinct, such that $g_1 \in K^l_{u_1}$ and $g_2 \in K^l_{u_2}$. We saw in the first part of this proof that we must have $\mathit{cross}\{R(e_1), R(e_2)\}$. This implies

*cross*$\{R(e_2), R(g_1)\}$ or *cross*$\{R(e_2), R(g_2)\}$. But both $\{e_2, g_1\} \in E$ or $\{e_2, g_2\} \in E$ together with $\{e_1, g_1\}, \{e_1, g_2\} \in E$ contradict Claim 6.3.6. We conclude that we must have the situation *cross*$\{R(e_1), R(d)\}$.

(ii) We assume the opposite for a contradiction. Let $u, u_1, u_2, u_3 \in U^l$ be pairwise distinct, and let $e_1, e_2, e_3 \in K_u^l$ be pairwise distinct. Let $g_i \in K_{u_i}^l$ such that $\{g_i, e_i\} \in E$, $i \in [3]$. By Item (i) we only have the cross situations *cross*$\{R(g_i), R(e_i)\}$, $i \in [3]$, *cross*$\{R(e_1), R(e_2)\}$, *cross*$\{R(e_1), R(e_3)\}$, and also *cross*$\{R(e_2), R(e_3)\}$. W. l. o. g. we can assume *cross*$(R(g_1), R(e_1))$. (Otherwise, consider the transpose of $M$.) Then we have *cross*$(R(e_2), R(e_1))$, because *cross*$(R(e_1), R(e_2))$ would imply $R(g_1) \cap R(e_2) \neq \emptyset$. But $\{g_1, e_2\} \in E$ is in contradiction to Claim 6.3.6. By analogous arguments we obtain *cross*$(R(e_2), R(g_2))$ and *cross*$(R(e_2), R(e_3))$. $B(R(e_1)) \cap B(R(e_3)) = \emptyset$ cannot be the case, because of $\{e_1, e_3\} \in E$ ($K_u^l$ is a clique). But $B(R(e_1)) \cap B(R(e_3)) \neq \emptyset$ implies $R(e_3) \cap R(g_1) \neq \emptyset$ and thus, $\{e_3, g_1\} \in E$, again contradicting Claim 6.3.6. We conclude that in each clique $K_u^l$ there are at most two nodes adjacent to other cliques.

(iii) We assume for a contradiction that the cliques $K_{u_i}^l$, $i \in [3]$, are non-trivial. Then there exist elements $e_i' := \{u_i, v_i'\} \in K_{u_i}^l$, $e_i' \neq e_i$, $v_i' \in U^r$, $i \in [3]$. There exists a node $v \in U^r$ such that $e_i = \{u_i, v\}$, because $\{e_i \mid i \in [3]\}$ forms a triangle in $G$. This set is a subset of the right clique $K_v^r$. In addition, we have $e_i' \in K_{v_i'}^r$, and $e_i, e_i'$ are adjacent, $i \in [3]$. Thus, $|B_v^r| \geq 3$ in contradiction to Item (ii).

(iv) Denote with $D$ the induced subgraph $G(\bigcup_{u \in F^l} B_u^l)$. We assume for a contradiction that $D$ is not bipartite. Then $D$ contains an odd cycle. As $G$ is beautiful, also $D$ is beautiful. One can show by induction on the cycle length that a Berge graph containing an odd cycle as a subgraph (not necessarily induced) contains a triangle. Thus, $D$ contains a triangle $\{e_1, e_2, e_3\}$. Each node $e_i$ must lie in a separate non-trivial clique by Claim 6.3.6. But this contradicts Item (iii). We conclude that $D$ must be bipartite. □

Definition 6.3.8. *We call a graph a* Path-or-Even-Cycle-of-Cliques, *if it consists of a path of arbitrary length or a cycle of even length $\geq 6$, where to each edge of the respective path or cycle there may be a clique attached. Those attached cliques are of arbitrary size, contain a single edge of the respective path or cycle, and contain only additional nodes and edges.*

See Figure 6.3.1 for an example.

Theorem 6.3.9. *Let $G$ be a graph as defined at the beginning of this section. If $G$ is beautiful, then $G$ is a Path-or-Even-Cycle-of-Cliques.*

Proof. Our strategy is to first delete some edges in $\tilde{G}$ to obtain $\tilde{G}'$. We note that the line graph $L(\tilde{G}')$ has a simple structure. Then we add the deleted edges to obtain the structure of $L(\tilde{G})$. From $G$ we delete all edges $e = \{u, v\}$, $u \in U^l$, $v \in U^r$, which are trivial cliques, i. e. $K_u^l = \{e\}$, or which are non-connection nodes in a non-trivial clique of $G$, i. e. $e \in K_u^l - B_u^l$. Note that in the first case we have $d_{\tilde{G}}(u) = 1$, while in the second we have $d_{\tilde{G}}(v) = 1$. We obtain a square-free bipartite graph $\tilde{G}'$, where all nodes have degree between one and two, because the only edges we left over are the connection nodes of $G$: By Lemma 6.3.7(ii) we have $|B_u^l| \leq 2$ implying $d_{\tilde{G}'}(u) \leq 2$ for $u \in U^l$, and by Lemma 6.3.7(iv) we have $G' := G(\bigcup_{u \in F^l} B_u^l)$ bipartite implying $d_{\tilde{G}'}(v) \leq 2$ for $v \in U^r$, because otherwise we would have a triangle in $G'$.

Thus, $L(\tilde{G}')$ is a path of arbitrary length or a cycle of even length $\geq 6$. Now, we add the deleted edges and distinguish three cases:

1. Let $K_u^l$ be a non-trivial clique with exactly one connection node $e$. In this case $e$ is the end node of the path $L(\tilde{G}')$. Adding the deleted edges from $K_u^l - B_u^l$ just adds a clique to the node $e$ in $L(\tilde{G}')$.

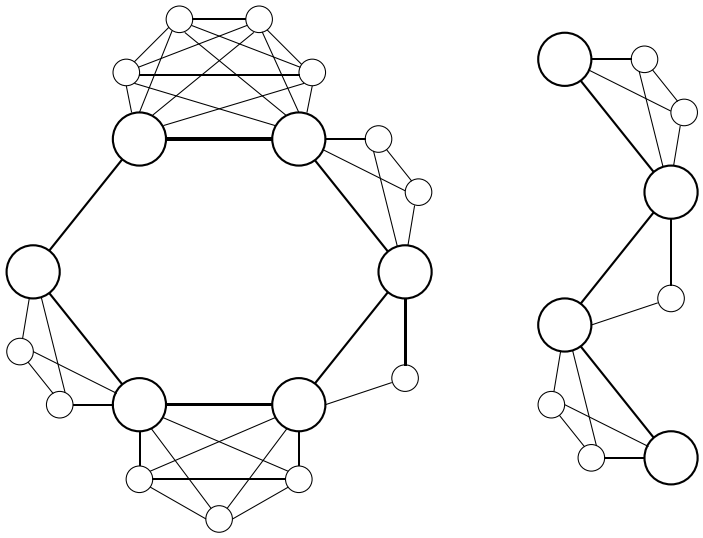

Figure 6.3.1: Path-or-Even-Cycle-of-Clique graphs

2. Let $K_u^l$ be a non-trivial clique with two connection nodes $e_1, e_2$. In this case $\{e_1, e_2\}$ is an edge of the path or cycle $L(\tilde{G}')$. Adding the deleted edges from $K_u^l - B_u^l$ just adds a clique to the edge $\{e_1, e_2\}$ in $L(\tilde{G}')$.

3. Now, we consider trivial cliques. We collect their edges in classes $M_v$, $v \in U^r$, defined by $M_v := \{e \mid e = \{u, v\}, K_u^l = \{e\} \text{ trivial }\}$. We fix a node $v$ and distinguish three cases according to the number of connection nodes $e \in \bigcup_{u \in F^l} B_u^l$ incident with $v$:
   3.1. In case there is no such edge $e$, the graph $L(G)$ is just a clique with nodes from $M_v$.
   3.2. In case there is a single edge $e$, adding the edges $M_v$ to $\tilde{G}'$ attaches a clique to the end node $e$ in $L(\tilde{G}')$.
   3.3. In case there are exactly two edges $e_1, e_2$, adding the edges $M_v$ to $\tilde{G}'$ attaches a clique to the edge $\{e_1, e_2\}$ in $L(\tilde{G}')$.

In all cases we obtain $L(G)$ from the path or even cycle $L(\tilde{G}')$ by adding at most one clique of arbitrary size to each edge or end node of $L(\tilde{G}')$. □

Lemma 6.3.10. *A Path-or-Even-Cycle-of-Cliques is a cs-graph.*

Proof. In case of a path of cliques, we generate the path by writing a "stair" matrix, where long columns of ones alternate with long rows of ones. Then, we generate the cliques by filling in the "corners" in the stairs.

In case of a cycle, we do the same, but we start with a long column, let the stair go down, and end with a long row. We select a clique $S$ and fill in all the cliques except $S$. This is possible, because the cycle is even. Let $m$ be the size of $S$. Now it remains to represent $S$ in the matrix. We add a triangular matrix $T_m$ such that the longest column of ones in $T_m$ (rightmost column) is below the first long column of the stair and such that the longest row of ones in $T_m$ (first row) is left to the last long row of the stair. □

The construction presented in the proof above is illustrated in Figure 6.3.2 for the cycle of cliques shown in Figure 6.3.1. The rectangles drawn correspond to the nodes of the cycle.

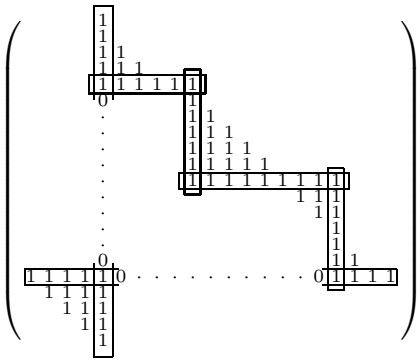

Figure 6.3.2: Representation of a cycle of cliques

We observe that every induced subgraph of a Path-or-Even-Cycle-of-Cliques is a Path-or-Even-Cycle-of-Cliques. Thus, by Lemma 6.3.10 we have

THEOREM 6.3.11. *A Path-or-Even-Cycle-of-Cliques is beautiful.* □

By Theorems 6.3.9 and 6.3.11 we finally obtain

COROLLARY 6.3.12. *A (connected) line graph of a square-free bipartite graph is beautiful iff it is a Path-or-Even-Cycle-of-Cliques.*

**Alternative proof.**

We present an elegant alternative proof of Theorem 6.3.9 based on a proof sketch suggested by an anonymous referee of Wunderlich (2009a). In contrast to Lemma 6.3.7 the new proof avoids considering rectangle situations by exploiting in a clever way the fact that stars are forbidden. We first have to define asteroids and caterpillars:

DEFINITION 6.3.13 (Asteroid). *An* asteroid *is a graph whose line graph is the star.*

Thus, an asteroid is a tree on seven nodes obtained by subdividing once every edge of a *claw*, where the claw is the graph $(\{a, b, c, d\}, \{ab, ac, ad\})$.

DEFINITION 6.3.14 (Caterpillar). *A* caterpillar *is a graph such that if all nodes of degree one and their incident edges are removed, the remainder of the graph (the* spine*) forms a path of arbitrary length or an even cycle of length at least six.*
*In the first case, we call the graph a* path caterpillar*, in the latter a* cycle caterpillar*.*
*The nodes of degree one are called* leg nodes*, the other ones* spine nodes*.*
*If $l$ is a leg node, then we denote with $s(l)$ its unique neighbor.*

ALTERNATIVE PROOF OF THEOREM 6.3.9. Let $G = L(\tilde{G})$ be a beautiful connected line graph of a square-free bipartite graph $\tilde{G}$. We prove that $G$ is the line graph of a caterpillar, and thus is a Path-or-Even-Cycle-of-Cliques.

*Claim 1: $\tilde{G}$ does not contain an asteroid as a (not necessarily induced) subgraph.* Otherwise, $G$ contains a star as an induced subgraph in contradiction to Theorem 6.2.6.

Let $C$ be a (not necessarily induced) subgraph of $\tilde{G}$ that is a caterpillar such that $|V(C)|$ is maximal. Among all $C$'s, we take one with maximal $|E(C)|$.

*Claim 2: No node $v$ of $\tilde{G}$ is adjacent to a leg node of $C$.* We assume the opposite for a contradiction. Let $v$ be adjacent to a leg node $l$ in $C$. We define the graph $C'$ as

$C$, where we have added $v$ and the edge $\{v, l\}$. If there exist nodes $a, b, d, e$ in $C$ such that $a, b, s(l), d, e$ forms a path in $C$, then $C$ contains an asteroid in contradiction to Claim 1. Otherwise, $C'$ is a path caterpillar contradicting the maximality of $C$.

*Claim 3:* $V(C) = V(\tilde{G})$. We assume the opposite for a contradiction. Let $v$ be a node in $\tilde{G}$ that is not in $C$. Since $\tilde{G}$ is connected there exists a node $s$ in $C$ adjacent to $v$. We define the graph $C'$ as $C$, where we have added $v$ and the edge $\{v, s\}$. By Claim 2, $v$ is connected to a spine node $s$ of $C$. Thus, $C'$ is a caterpillar contradicting the maximality of $C$.

*Claim 4:* $E(C) = E(\tilde{G})$. We assume the opposite for a contradiction. Let $e := \{v_1, v_2\}$ be an edge in $\tilde{G}$ that is not in $C$. Let the natural number $\delta$ be the distance between $v_1$ and $v_2$ in $C$. This is defined, because $C$ is connected. The graph $\tilde{G}$ contains a cycle $C_{\delta+1}$ as a (not necessarily induced) subgraph. Thus, if $2 \leq \delta \leq 4$ then $\tilde{G}$ contains a $C_k$ as an induced subgraph for $3 \leq k \leq 5$. But $k = 3$ or $k = 5$ is in contradiction to $\tilde{G}$ bipartite, and $k = 4$ is in contradiction to $\tilde{G}$ square-free. We assume that $\delta \geq 5$ and distinguish several cases:

1. We assume that $C$ is a path caterpillar.
   - 1.1. We assume that $v_1$ and $v_2$ are leg nodes.
     - 1.1.1. If $s(v_1)$ and $s(v_2)$ are end points of the spine of $C$, then $C + e$ is a cycle caterpillar contradicting the maximality of $|E(C)|$.
     - 1.1.2. If $s(v_1)$ is not an end point of the spine of $C$, then because of $\delta \geq 5$ there exist nodes $u, v, x, y$ in $C$ distinct from $v_1, v_2, s(v_1), s(v_2)$ such that $v, s(v_1), x, y$ is a path on the spine of $C$ and $u$ is adjacent to $v$. The node $u$ may be a spine or a leg node in $C$. Thus,

       $$(\{u, v, x, y, v_1, v_2, s(v_1)\}, \{uv, vs(v_1), s(v_1)x, xy, s(v_1)v_1, v_1v_2\})$$

       is an asteroid in $\tilde{G}$ contradicting Claim 1.
     - 1.1.3. Analogously, if $s(v_2)$ is not an end point of the spine of $C$.
   - 1.2. We assume that $v_1$ is a spine node of $C$.
     - 1.2.1. If $v_1$ is not an end point of the spine of $C$, then because of $\delta \geq 5$ there exist nodes $u, v, x, y$ in $C$ distinct from $v_1, v_2$ such that $v, v_1, x, y$ is a path on the spine of $C$ and $u$ is adjacent to $v$. The node $u$ may be a spine or a leg node in $C$. Let $z \neq v_1$ be a neighbor of $v_2$ on the spine of $C$. Thus,

       $$(\{u, v, x, y, z, v_1, v_2\}, \{uv, vv_1, v_1x, xy, v_1v_2, v_2z\})$$

       is an asteroid in $\tilde{G}$ contradicting Claim 1.
     - 1.2.2. If $v_1$ is an end point of the spine of $C$, then we have to distinguish three more cases:
       - 1.2.2.1. If $v_2$ is an end point of the spine of $C$, then $C+e$ is a cycle caterpillar contradicting the maximality of $|E(C)|$.
       - 1.2.2.2. If $v_2$ is on the spine of $C$ but not an end point, then because of $\delta \geq 5$ there exist nodes $u, v, x, y$ in $C$ distinct from $v_1, v_2$ such that $u, v, v_2, x$ is a path on the spine of $C$ and $y$ is adjacent to $x$. The node $y$ may be a spine or a leg node in $C$. Let $z \neq v_2$ be a neighbor of $v_1$ on the spine of $C$. Thus,

         $$(\{u, v, x, y, z, v_1, v_2\}, \{uv, vv_2, v_2x, xy, v_1v_2, v_1z\})$$

         is an asteroid in $\tilde{G}$ contradicting Claim 1.
       - 1.2.2.3. We assume that $v_2$ is a leg node in $C$. If $s(v_2)$ is an end point of the spine of $C$, then $C + e$ is a cycle caterpillar contradicting the maximality of $|E(C)|$. Otherwise, because of $\delta \geq 5$ there exist

nodes $u, v, x, y$ in $C$ distinct from $v_1, v_2, s(v_2)$ such that $u, v, s(v_2), x$ is a path on the spine of $C$ and $y$ is adjacent to $x$. The node $y$ may be a spine or a leg node in $C$. Thus,

$$(\{u, v, x, y, v_1, v_2, s(v_2)\}, \{uv, vs(v_2), s(v_2)x, xy, s(v_2)v_2, v_1v_2\})$$

is an asteroid in $\tilde{G}$ contradicting Claim 1.

1.3. Analogously, if $v_2$ is a spine node of $C$.

2. We assume that $C$ is a cycle caterpillar.

2.1. If $v_1$ is a leg node, then because of $\delta \geq 5$ there exist nodes $u, v, x, y$ distinct from $v_1, v_2, s(v_1)$ such that $u, v, s(v_1), x, y$ is a path on the spine of $C$. Thus,

$$(\{u, v, x, y, s(v_1), v_1, v_2\}, \{uv, vs(v_1), s(v_1)x, xy, s(v_1)v_1, v_1v_2\})$$

is an asteroid in $\tilde{G}$.

2.2. Analogously, if $v_2$ is a leg node.

2.3. If $v_1$ and $v_2$ are spine nodes, then because of $\delta \geq 5$ there exist nodes $u, v, x, y$ distinct from $v_1, v_2$ such that $u, v, v_1, x, y$ is a path on the spine of $C$. Let $z \neq v_1$ be a neighbor of $v_2$ on the spine of $C$. Thus,

$$(\{u, v, x, y, z, v_1, v_2\}, \{uv, vv_1, v_1x, xy, v_1v_2, v_2z\})$$

is an asteroid in $\tilde{G}$.

In all cases, we have a contradiction to Claim 1.

From Claims 3 and 4 we deduce that $G$ is the line graph of a caterpillar. □

## 6.4 Concluding remarks

In summary, we proved that not every graph is a cs-graph, and we characterized beautiful square-free bipartite graphs and beautiful line graphs of square-free bipartite graphs. Certainly, these findings show a need for further research, because many open questions still remain. We list some of the most urgent: In Theorem 6.2.6 a list of non-cs-graphs was given. Is this list complete? If not, can the list of non-cs-graphs at least be described by a finite number of well-known graph families? Is it true that every graph is the induced subgraph of a cs-graph? Does there exist a characterization or decomposition theorem for beautiful graphs, e. g. in the spirit of Conforti *et al.* (2004)? If $K_4$s are forbidden, i. e. we do not allow spade situations, what can be said about such cs- and beautiful graphs? How many cuts are necessary to break up a matrix into submatrices such that each submatrix represents a beautiful graph? Finally, it would be very interesting to find applications involving cs- or beautiful graphs in fields other than communication complexity.

# Bibliography


Rudolf Ahlswede & Ning Cai (1994). On communication complexity of vector-valued functions. *IEEE Transactions on Information Theory* **40**(6), 2062–2067.

Alfred V. Aho, Jeffrey D. Ullman & Mihalis Yannakakis (1983). On Notions of Information Transfer in VLSI Circuits. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing, 25–27 April 1983, Massachusetts, USA*, 133–139. ACM.

Noga Alon & Eyal Lubetzky (2006). The Shannon capacity of a graph and the independence numbers of its powers. *IEEE Transactions on Information Theory* **52**(5), 2172–2176.

Dana Angluin (1980). On Counting Problems and the Polynomial-Time Hierarchy. *Theor. Comput. Sci.* **12**, 161–173.

László Babai, Peter Frankl & Janos Simon (1986). Complexity classes in communication complexity theory (preliminary version). In *27th Annual Symposium on Foundations of Computer Science, FOCS 1986, 27–29 October 1986, Toronto, Ontario, Canada*, 337–347. IEEE Computer Society.

José L. Balcázar, Josep Díaz & Joaquim Gabarró (1990). *Structural Complexity II.* Texts in Theoretical Computer Science, An EATCS Series. Springer-Verlag, 1st edition.

José L. Balcázar, Josep Díaz & Joaquim Gabarró (1995). *Structural Complexity I.* Texts in Theoretical Computer Science, An EATCS Series. Springer-Verlag, 2nd edition.

Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar & D. Sivakumar (2004). An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.* **68**(4), 702–732.

Paul Beame, Toniann Pitassi, Nathan Segerlind & Avi Wigderson (2006). A Strong Direct Product Theorem for Corruption and the Multiparty Communication Complexity of Disjointness. *Computational Complexity* **15**(4), 391–432.

Claude Berge (1961). Färbung von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind (Zusammenfassung). *Wissenschaftliche Zeitschrift, Martin Luther Universität Halle-Wittenberg, Mathematisch-Naturwissenschaftliche Reihe* 114–115.

Harry Buhrman, Tao Jiang, Ming Li & Paul M. B. Vitányi (2000). New applications of the incompressibility method: Part II. *Theor. Comp. Sci.* **235**(1), 59–70.

Harry Buhrman, Nikolai K. Vereshchagin & Ronald de Wolf (2007). On Computation and Communication with Small Bias. In *22nd Annual IEEE Conference on Computational Complexity (CCC 2007), 13–16 June 2007, San Diego, California, USA*, 24–32. IEEE Computer Society.

Amit Chakrabarti, Yaoyun Shi, Anthony Wirth & Andrew Chi-Chih Yao (2001). Informational Complexity and the Direct Sum Problem for Simultaneous Message Complexity. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, October 14–17, Las Vegas Nevada, USA*, 270–278. IEEE Society.

Mahdi Cheraghchi (2005). On Matrix Rigidity and the Complexity of Linear Forms. *Electronic Colloquium on Computational Complexity (ECCC)* (070).

Benny Chor & Oded Goldreich (1988). Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity. *SIAM J. Comput.* **17**(2), 230–261.

Maria Chudnovsky, Neil Robertson, Paul Seymour & Robin Thomas (2006). The Strong Perfect Graph Theorem. *Annals of Mathematics* **164**, 51–229.

Fan Chung & Ronald Graham (2002). Sparse quasi-random graphs. *Combinatorica* **22**, 217–244.

Fan R. K. Chung, Ronald L. Graham & Richard M. Wilson (1989). Quasi-random graphs. *Combinatorica* **9**(4), 345–362.

Fan R. K. Chung & Prasad Tetali (1993). Communication Complexity and Quasi Randomness. *SIAM J. Discrete Math.* **6**(1), 110–123.

Bruno Codenotti (2000). Matrix rigidity. *Linear Algebra and its Applications* **304**(1–3), 181–192.

Bruno Codenotti, Pavel Pudlák & Giovanni Resta (2000). Some structural properties of low-rank matrices related to computational complexity. *Theor. Comput. Sci.* **235**(1), 89–107.

Michele Conforti, Gérard Cornuéjols & Kristina Vušković (2004). Square-free perfect graphs. *J. Comb. Theory, Ser. B* **90**(2), 257–307.

David Conlon (2008). A new upper bound for the bipartite Ramsey problem. *Journal of Graph Theory* **58**(4), 351–356.

Thomas M. Cover & Joy A. Thomas (1991). *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley & Sons Inc.

Carsten Damm, Matthias Krause, Christoph Meinel & Stephan Waack (2004). On relations between counting communication complexity classes. *J. Comput. Syst. Sci.* **69**(2), 259–280.

Reinhard Diestel (2005). *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, 3rd edition.

Martin Dietzfelbinger & Henning Wunderlich (2007). A characterization of average case communication complexity. *Inf. Process. Lett.* **101**(6), 245–249.

Ding-Zhu Du & Ker-I Ko (2000). *Theory of Computational Complexity*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., 1st edition.

Jürgen Forster (2002). A linear lower bound on the unbounded error probabilistic communication complexity. *J. Comput. Syst. Sci.* **65**(4), 612–625.

Lance Fortnow (1997). *Counting Complexity*, 81–107. In Selman & Hemaspaandra (1997).

Joel Friedman (1993). A note on matrix rigidity. *Combinatorica* **13**(2), 235–239.

Bernd Halstenberg & Rüdiger Reischuk (1990). Relations between Communication Complexity Classes. *J. Comput. Syst. Sci.* **41**(3), 402–429.

Prahladh Harsha, Rahul Jain, David A. McAllester & Jaikumar Radhakrishnan (2007). The Communication Complexity of Correlation. In *22nd Annual IEEE Conference on Computational Complexity (CCC 2007), 13–16 June 2007, San Diego, California, USA*, 10–23. IEEE Computer Society.

Lane A. Hemaspaandra & Mitsunori Ogihara (2002). *The Complexity Theory Companion*. Texts in Theoretical Computer Science, An EATCS Series. Springer-Verlag.

Shlomo Hoory, Nathan Linial & Avi Wigderson (2006). Expander Graphs and Applications. *Bull. Amer. Math Soc.* **43**, 439–561.

Stefan Hougardy (2006). Classes of Perfect Graphs. *Discrete Mathematics* **306**(19–20), 2529–2571.

Juraj Hromkovic (2000). *Communication Complexity and Parallel Computing*. Texts in Theoretical Computer Science – An EATCS Series. Springer-Verlag.

Rahul Jain, Jaikumar Radhakrishnan & Pranab Sen (2003). A Direct Sum Theorem in Communication Complexity via Message Compression. In *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, Eindhoven, The Netherlands, June 30 – July 4, 2003. Proceedings*, Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow & Gerhard J. Woeginger, editors, volume 2719 of *Lecture Notes in Computer Science*, 300–315. Springer-Verlag.

T. S. Jayram, Ravi Kumar & D. Sivakumar (2003). Two applications of information complexity. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9–11, 2003, San Diego, CA, USA*, 673–682. ACM.

Bala Kalyanasundaram & Georg Schnitger (1992). The Probabilistic Communication Complexity of Set Intersection. *SIAM J. Discrete Math.* **5**(4), 545–557.

Mauricio Karchmer & Avi Wigderson (1990). Monotone Circuits for Connectivity Require Super-Logarithmic Depth. *SIAM J. Discrete Math.* **3**(2), 255–265.

Hartmut Klauck (2001). Lower Bounds for Quantum Communication Complexity. In *42nd Annual Symposium on Foundations of Computer Science, October 14–17, Las Vegas Nevada, USA*, 288–297. IEEE.

Hartmut Klauck (2003). Rectangle Size Bounds and Threshold Covers in Communication Complexity. In *18th Annual IEEE Conference on Computational Complexity,7–10 July 2003, Aarhus, Denmark*, 118–134. IEEE Computer Society.

Johannes Köbler, Uwe Schöning & Jacobo Torán (1993). *The Graph Isomorphism Problem – Its Structural Complexity*. Birkhäuser Boston.

Michael Krivelevich & Benny Sudakov (2006). Pseudo-random graphs. In *More sets, graphs and numbers*, E. Győri, G. O. H. Katona & Laszlo Lovász, editors, volume 15 of *Bolyai Soc. Math. Studies*, 199–262. Springer-Verlag.

Eyal Kushilevitz & Noam Nisan (1997). *Communication Complexity*. Cambridge University Press.

Nati Linial & Adi Shraibman (2007). Lower bounds in communication complexity based on factorization norms. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11–13, 2007*, David S. Johnson & Uriel Feige, editors, 699–708. ACM.

Satyanarayana V. Lokam (2000). On the rigidity of Vandermonde matrices. *Theor. Comput. Sci.* **237**(1–2), 477–483.

Satyanarayana V. Lokam (2001). Spectral Methods for Matrix Rigidity with Applications to Size-Depth Trade-offs and Communication Complexity. *J. Comput. Syst. Sci.* **63**(3), 449–473.

Satyanarayana V. Lokam (2006). Quadratic Lower Bounds on Matrix Rigidity. In *Theory and Applications of Models of Computation, Third International Conference, TAMC 2006, Beijing, China, May 15–20, 2006, Proceedings*, Jin-Yi Cai, S. Barry Cooper & Angsheng Li, editors, volume 3959 of *Lecture Notes in Computer Science*, 295–307. Springer-Verlag.

László Lovász (1979). On the Shannon capacity of a graph. *IEEE Transactions on Information Theory* **25**, 1–7.

Terry A. McKee & Fred R. McMorris (1999). *Topics in Intersection Graph Theory*. SIAM Monographs on Discrete Mathematics and Applications.

Kurt Mehlhorn & Erik Meineche Schmidt (1982). Las Vegas Is better than Determinism in VLSI and Distributed Computing (Extended Abstract). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing, 5–7 May 1982, San Francisco, California, USA*, 330–337. ACM.

Gatis Midrijanis (2005). Three lines proof of the lower bound for the matrix rigidity. *CoRR* **abs/cs/0506081**.

Rajeev Motwani & Prabhakar Raghavan (1995). *Randomized Algorithms*. Cambridge University Press.

Ilan Newman (1991). Private vs. Common Random Bits in Communication Complexity. *Inf. Process. Lett.* **39**(2), 67–71.

Noam Nisan & Avi Wigderson (1995). On Rank vs. Communication Complexity. *Combinatorica* **15**(4), 557–565.

Alon Orlitsky & Abbas El Gamal (1990). Average and randomized communication complexity. *IEEE Transactions on Information Theory* **36**(1), 3–16.

Christos H. Papadimitriou & Stathis Zachos (1983). Two remarks on the power of counting. In *Theoretical Computer Science, 6th GI-Conference, Dortmund, Germany, January 5–7, 1983, Proceedings*, Armin B. Cremers & Hans-Peter Kriegel, editors, volume 145 of *Lecture Notes in Computer Science*, 269–276. Springer-Verlag.

Pavel Pudlák (1994). Communication in Bounded Depth Circuits. *Combinatorica* **14**(2), 203–216.

Pavel Pudlák & Vojtech Rödl (1994). Some combinatorial-algebraic problems from complexity theory. *Discrete Mathematics* **136**(1–3), 253–279.

Jorge L. Ramírez-Alfonsín & Bruce A. Reed (2001). *Perfect Graphs*. Wiley-Intersciences Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc.

Ran Raz (1995). Fourier Analysis for Probabilistic Communication Complexity. *Computational Complexity* **5**(3/4), 205–221.

Ran Raz & Boris Spieker (1993). On the "log rank"-Conjecture in Communication Complexity. In *34th Annual Symposium on Foundations of Computer Science, FOCS 1993, 3–5 November 1993, Palo Alto, California, USA*, 168–176. IEEE Computer Society.

Alexander A. Razborov & Alexander A. Sherstov (2008). The Sign-Rank of $AC^O$. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25–28, 2008, Philadelphia, PA, USA*, 57–66. IEEE Computer Society.

Uwe Schöning (1986). *Complexity and Structure*, volume 211 of *Lecture Notes in Computer Science*. Springer-Verlag.

Uwe Schöning (1988). The power of counting. In Selman (1988), 204–223.

Uwe Schöning (1989). Probabilistic Complexity Classes and Lowness. *J. Comput. Syst. Sci.* **39**(1), 84–100.

Uwe Schöning (1991). Recent Highlights in Structural Complexity Theory (invited talk). In *SOFSEM'91, Nizké Tratry (CSFR)*, Conference Proceedings, 205–216. Springer-Verlag.

Alan L. Selman (editor) (1988). *Complexity Theory Retrospective*. Foundations of Computing. Springer-Verlag.

Alan L. Selman & Lane A. Hemaspaandra (editors) (1997). *Complexity Theory Retrospective II*. Foundations of Computing. Springer-Verlag.

Claude Elwood Shannon (1948). A mathematical theory of communication. *Bell Sys. Tech. Journal* **27**, 379–423 and 623–656.

Claude Elwood Shannon (1965). The zero-error capacity of a noisy channel. *IRE Transactions on Information Theory* **2**, 8–19.

Mohammad Amin Shokrollahi, Daniel A. Spielman & Volker Stemann (1997). A Remark on Matrix Rigidity. *Inf. Process. Lett.* **64**(6), 283–285.

Seinosuke Toda (1991). PP is as Hard as the Polynomial-Time Hierarchy. *SIAM J. Comput.* **20**(5), 865–877.

Leslie G. Valiant (1977). Graph-Theoretic Arguments in Low-Level Complexity. In *Mathematical Foundations of Computer Science 1977, 6th Symposium, Tatranska Lomnica, Czechoslovakia, September 5–9, 1977, Proceedings*, Jozef Gruska, editor, volume 53 of *Lecture Notes in Computer Science*, 162–176. Springer-Verlag.

Leslie G. Valiant & Vijay V. Vazirani (1986). NP is as Easy as Detecting Unique Solutions. *Theor. Comput. Sci.* **47**(3), 85–93.

Umesh V. Vazirani (1987). Strong communication complexity or generating quasirandom sequences form two communicating semi-random sources. *Combinatorica* **7**(4), 375–392.

Ronald de Wolf (2006). Lower Bounds on Matrix Rigidity Via a Quantum Argument. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10–14, 2006, Proceedings, Part I*, Michele Bugliesi, Bart Preneel, Vladimiro Sassone & Ingo Wegener, editors, volume 4051 of *Lecture Notes in Computer Science*, 62–71. Springer-Verlag.

Henning Wunderlich (2009a). On cover-structure graphs. *Discrete Applied Mathematics* **157**(15), 3289–3299.

Henning Wunderlich (2009b). On Toda's Theorem in Structural Communication Complexity. In *SOFSEM 2009: Theory and Practice of Computer Science, 35th Conference on Current Trends in Theory and Practice of Computer Science, Spindleruv Mlýn, Czech Republic, January 24–30, 2009. Proceedings*, Mogens Nielsen, Antonín Kucera, Peter Bro Miltersen, Catuscia Palamidessi, Petr Tuma & Frank D. Valencia, editors, volume 5404 of *Lecture Notes in Computer Science*, 609–620. Springer-Verlag.

Andrew Chi-Chih Yao (1979). Some Complexity Questions Related to Distributive Computing (Preliminary Report). In *Conference Record of the Eleventh Annual ACM Symposium on Theory of Computing, 30 April–2 May, 1979, Atlanta, Georgia, USA*, 209–213. ACM.

Andrew Chi-Chih Yao (1983). Lower Bounds by Probabilistic Arguments (Extended Abstract). In *24th Annual Symposium on Foundations of Computer Science, FOCS 1983, 7–9 November 1983, Tucson, Arizona, USA*, 420–428. IEEE Computer Society.

# Erklärung gemäß Anlage 1 der Promotionsordnung

Ich versichere, daß ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Weitere Personen waren an der inhaltlich-materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater oder anderer Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer Prüfungsbehörde vorgelegt.

Ich bin darauf hingewiesen worden, daß die Unrichtigkeit der vorstehenden Erklärung als Täuschungsversuch angesehen wird und den erfolglosen Abbruch des Promotionsverfahrens zur Folge hat.

(Ort, Datum) (Unterschrift)

# Thesen

Die in den Thesen genannten Komplexitätsklassen beziehen sich alle auf Klassen aus der Kommunikationskomplexität.

1. Für jede Funktion und jede Verteilung auf den Eingaben stimmt die durchschnittliche deterministische Informationskomplexität bis auf einen konstanten Faktor mit der durchschnittlichen deterministischen Kommunikationskomplexität überein.

2. Die durchschnittliche deterministische Informationskomplexität liefert mittels der Rechtecksgrößenmethode untere Schranken für die randomisierte public coin Las Vegas Kommunikationskomplexität, die um einen konstanten Faktor besser sind, als die bisher bekannten.

3. Die Sätze von Toda gelten in Yaos Kommunikationsmodell.

4. Wir entwickeln eine neues Maß, den approximativen $\mathbb{F}_2$-Rang, und zeigen, dass dieser die BP-Parität-P-Komplexität charakterisiert.

5. Es besteht eine enge Beziehung zwischen einer Booleschen Variante des Konzepts „matrix rigidity“ und dem approximativen $\mathbb{F}_2$-Rang. Daraus ergibt sich ein Maßkonzentrationsresultat für die BP-Parität-P-Komplexität: die meisten Funktionen haben eine BP-Parität-P-Komplexität von $\Omega(n/\log n)$.

6. Wir entwickeln ein Protokoll für die innere Produktfunktion mod 2 mit wenigen Alternierungen. Dies könnte darauf hinweisen, dass die Klassen BP-Parität-P und Polynomieller Platz verschieden sind.

7. Wir beweisen, dass Adjazenzprobleme von dünnen quasi-zufälligen Graphfamilien eine hohe Parität-P-Komplexität besitzen.

8. Wir definieren und untersuchen die Klasse der Überdeckungsstrukturgraphen und zeigen für mehrere Graphen, insbesondere für Quadrate und ungerade Löcher, dass diese keine Überdeckungsstrukturgraphen sind.

9. Schöne Graphen haben die Eigenschaft, dass jeder induzierte Untergraph ein Überdeckungsstrukturgraph ist. Wir untersuchen diese sehr spezielle Klasse quadratfreier Berge-Graphen und zeigen, dass jeder quadratfreie bipartite Graph schön ist, und dass die schönen Kantengraphen quadratfreier bipartiter Graphen genau diejenigen Graphen sind, die aus Wegen beliebiger und Kreisen gerader Länge bestehen, an deren Kanten Cliquen beliebiger Größe angeheftet sein können.